APPROVAL PAGE FOR GRADUATE THESIS OR PROJECT

SUBMITTED IN PARTIAL FULLFILMENT OF REQUIREMENTS FOR DEGREE OF MASTER OF SCIENCE AT CALIFORNIA STATE UNIVERSITY, LOS ANGELES BY

MOHAMMED SHARIF SALEH CANDIDATE
COMPUTER SCIENCE FIELD OF CONCENTRATION

TITLE: PAIR-WISE KEY ESTABLISHMENT FOR WIRELESS SENSOR NETWORK

APPROVED: Dr. HUIPING GUO
Faculty Member
Signature

Dr. BEHZAD PARVIZ
Faculty Member
Signature

Dr. RAJ S. PAMULA
Department Chairperson
Signature

DATE: 08/24/2007
PAIR-WISE KEY ESTABLISHMENT FOR
WIRELESS SENSOR NETWORK

A Thesis
Presented to
The Faculty of the Department of Computer Science
California State University, Los Angeles

In Partial Fulfillment
Of the Requirements for the Degree
Master of Science

By
Mohammed Sharif Saleh
August 2007
ACKNOWLEDGEMENT

I would like to take this opportunity to thank the great and wonderful people of the computer science department at California State University, Los angels. I would like to give a special thanks to my great advisor and professor Dr. Huiping Guo, who without her help, support and encouragement this thesis would have not been possible. I also would like to thank Dr. Abbott who helped me out of widening my horizon of thinking and understanding, and the ability to plan and structure this thesis. Also I would like to thank all my great professors through the past year and half I attended the school, without them this experience would have not been interesting. I wish to thank my friends and colleagues at the school for their respect, help and encouragement to succeed and have this wonderful experience. Also I would like to thank my brothers, Jawad, Jihad, Ayman, and Ahmad for their love support and being their whenever I needed them, thank you guys for everything.

Finally, I would like to give a very special and warm thanks to two wonderful and important people in my life who with out them I may not be where I’m at right now, to my Mom (Hanieh Arafat) and Dad (Sharif Saleh) thank you for everything that you have gave me in my life from support, encouragement, moral, discipline, respect and standing beside me through the good and rough times, with out you life is meaningless, I love you guys.
ABSTRACT

Pair-Wise Key Establishment for Wireless Sensor Network

By

Mohammed Sharif Saleh

Managing security measures is a critical requirement, particularly in the application of sensor networks. In order to improve security in wireless sensor networks (WSN) it is necessary to encrypt messages before transmission. Point-to-point communication uses and end-to-end cryptography that achieves a high level of security. This is accomplished by distributing keys among all end points. The issue then becomes incompatibility with passive participation of the local broadcast. Close interaction of the nodes, short distances between them, energy constrains, and the unattended deployment of the system make WSN highly vulnerable to node capture. This paper attempts to establish a new pair-wise key schema that addresses the initialization process. The new model is loosely based on an existing model known as SHELL. We are proposing a pair-wise key establishment of neighboring sensor nodes that will improve the efficiency of the network architecture and reduces vulnerability to attacks and compromises. Our schema can be effectively applied to provide protection for the communication of neighboring nodes in the network; sensor nodes are capable of providing the same level of security as before while reducing the consequences of node compromise.
# TABLE OF CONTENTS

Acknowledgements...........................................................................................................III

Abstract..............................................................................................................................IV

List of Tables.....................................................................................................................VIII

List of Figures....................................................................................................................IX

Chapter

1. Introduction....................................................................................................................1

2. Basic Concepts.............................................................................................................6

   2.1 Introduction.............................................................................................................6

   2.2 Public Key Cryptography......................................................................................7

      2.2.1 RSA.............................................................................................................8

      2.2.2 Diffie-Hellman Key Agreement Protocol.....................................................9

   2.3 Secret Key Cryptography....................................................................................10

   2.4 Summary.............................................................................................................12


   3.1 Introduction.........................................................................................................14

   3.2 Wireless Sensor Network Architecture.........................................................15

      3.2.1 Hierarchical Sensor Networks...............................................................19

      3.2.2 Distributed Wireless Sensor Networks...............................................19

   3.3 Wireless Sensor Network Security.................................................................20

      3.3.1 Obstacles of WSN Security.................................................................21
3.3.2 Security Requirements

3.3.3 WSN Attacks

3.4 Defending Against Attacks

3.5 Group Key Establishment Protocols

3.5.1 Group Key Management

3.5.2 Tree-Based Key Management Protocols

3.6 Related Work

3.7 Summary

4. Pair-Wise Key Establishment

4.1 Threat Model

4.2 Network Model

4.3 Network Architecture

4.3.1 Command node

4.3.2 Gateways

4.3.3 Sensors

4.4 Network Bootstrapping

4.5 Pair-Wise Key Establishment

4.5.1 Key Distribution

4.5.2 Network Connection

4.6 Protecting Against Attack and Failure

4.6.1 Node (N1) Compromise
4.6.2 Node (N2) Compromise ........................................... 50

4.7 Summary ............................................................. 51

5. Conclusion ............................................................. 52

Bibliography ............................................................. 54
LIST OF TABLES

1. DES, 3DES, AES Table .................................................................................. 11

2. Hardware of Mica2 sensor node ................................................................. 15
LIST OF FIGURES

1. Sensor network architecture.................................................................17
2. WSN architecture.............................................................................20
3. Tree-Key based management protocol.............................................29
4. Network Architecture........................................................................41
CHAPTER 1

INTRODUCTION

As the name suggests, the wireless sensor network (WSN) is a collection of autonomous wireless devices that are linked together to form a single network able of capturing discrete sets of information at multiple locations over time. Each of the wireless devices is equipped with a unique sensor (depending on the specific needs of the end-user) that can be manipulated to detect certain types of activity or information. WSN can be used in a wide variety of applications. The earliest WSNs were pioneered for military purposes. Devices in the network were installed in multiple locations throughout a battle area thus permitting military decision-makers to evaluate conditions on the field and make critical decisions about changes in strategy in real-time. Not before long that the effectiveness of WSN and its ability to capture multiple data sets simultaneously were recognized and subsequently adapted for use in other industries including but not limited to monitoring environmental conditions, improving various regulatory agencies’ capacities to enhance security in a variety of contexts, and managing massive amounts of data and input in a streamlined and efficient fashion, such as in air traffic control settings.

The characteristics of WSN as described above make the application particularly effective for settings and situations in which security monitoring is needed. Although the strengths of the WSN in this settings cannot be denied, it is important to realize that the WSN does have some limitations. In order to optimize the potential and performance of WSN for any application, the nature of these limitations must be understood and
appropriate measures must be taken. This is especially important for cases that involve security management.

Three limitations characterize wireless sensor networks. (1) WSN’s generally require massive amounts of energy (and resources) in order to function properly. Therefore the source and cost of energy is a factor that must be taken into account when considering implementing or using a WSN. (2) The electronic architecture that is needed to maximize the efficacy of the WSN is considerably large. Running a WSN optimally requires vast computer resources. This can become an expensive endeavor and, in many cases, may require technical expertise. (3) Because WSN captures and transmits massive amounts of data, sufficient memory to store that information becomes an issue. This is of particular interest when data is being stored for long periods of time. The impact of each limitation on the end-user requires careful evaluation and planning.

Furthermore, there are other technology related security that can be categorized as: authenticity of information, availability of the system, confidentiality of the data, and integrity and quality of data obtained.

**Authenticity of information:** To understand the security concerns of WSN, lets consider an active military combat operation. For all of its strengths and ability to capture data, the WSN is not able to verify that the information or data it collects is authentic. It appears as though the technology behind WSN is not able to evaluate, discern, or otherwise separate data into discrete packets of authentic or inauthentic data. For example, the motion capture of WSN on battlefield
perimeter cannot in itself confirm or contest whether the moving object is human or another source, friendly or unfriendly.

**Availability of the system**: Because WSNs are often used in dynamic settings, they are vulnerable to system threats and outages. In the example of the battlefield for instance, there are a number of factors that could diminish or cut off the availability of the system altogether, thereby prohibiting data collection and environmental monitoring. Such factors might include deliberate interference attempts by the enemy, changes in climatic or environmental conditions that obscure the device’s capturing capacity, and destruction of the device by exploding ordnance. For these reasons, the user must consider the implications of system outages, whether periodic or permanent, and make contingency plans to deal with such outages and their consequences.

**Confidentiality of data**: This is a major concern. The user must consider who has administrator access to the WSN, and what degree of authority these individuals will have over the WSN, protocols must be developed—and enforced—for data management, and regulations for data usage must be determined. Because the WSN has multiple devices collecting multiple streams of data, it is particularly vulnerable to misuse of data.

**Integrity and quality of data obtained**: As with the first security concern, the WSN itself is not capable of discerning which data is “good” and which is not. It
will be up to the user to determine what is needed, how it should be captured, and how their integrity can be ensured.

Node communication is truly critical to optimal functioning and security of the WSN, yet it is an area that typically represents persistent challenges for WSN system administrators. To this end, we offer a new strategy for successful WSN communication, the core feature of which is a lightweight scheme for devices within a network. The scheme relies upon establishing a new pair-wise key between two neighboring nodes. The result of the strategy proposed herein is to secure communications between the neighboring nodes within a network that is prone to collusion attack. The benefit to the system and the user is that the vulnerability of the WSN will be reduced substantially. In addition, the solution that is proposed here is sensitive to the limitations and needs of the WSN. The assignment of key management activities across the entire network distributes energy and memory demands more evenly, thereby reducing threats to and vulnerabilities of the WSN.

This thesis proceeds in a logical order to allow the reader to be acclimated and introduced to the central topics of our discussion. Chapter two will present an overview of the security strategies that have typically been used for the management of WSN; two of the strategies discussed are private/public key cryptography and Diffie-Hellman. In chapter three, we present the background of WSN and discuss different approaches to WSN security in greater depth and detail and set the stage for the theoretical and practical aspects of the proposed security strategy. Chapter four will describe the importance of
establishing pair-wise key and how it should be applied. Finally, chapter five will provide a summary and conclusion of the materials presented.
CHAPTER 2
BASIC CONCEPTS

In this chapter, we are exploring the existing security implementations, such as encryption, cryptanalysis, and their different characteristics. Encryption algorithms alone do not solve all our encryption needs. Other security measures must be explored in order to prevent an attack from occurring in the first place. In the next few sections, we will discuss public key cryptography, followed by RSA and Secret key cryptography.

2.1 Introduction

A security attack is a hostile action that compromises the security of information. Different security mechanisms have been developed to detect or prevent security attacks. There are different types of security attacks: interruption, interception, modification, and fabrication. The most common state of art technology used in WSN is key base encryption protocol or cryptographic protocol. There are two algorithms for key-based encryption: public key also known as asymmetric and secret or private key also known as symmetric.

Cryptography is the ancient science of encoding messages such that only the sender and recipient of the messages can understand their content. Thanks to modern computers and their ability to perform complex mathematical operations in a fraction of the time needed by a human being, cryptography is now available to everyone. Cryptography is used to secure telephone, Internet, and email communications and to protect software or other digital properties.
A cryptographic algorithm or cipher algorithm is a function or cryptographic system that is used to perform encryption or decryption. Before data is encrypted it is considered to be plaintext and can be read by anyone. After it has been encrypted, it is known as ciphertext that is not readable by third parties. Usually algorithms are a set of well defined instructions for a task, which in this case is to encrypt or decrypt data.

Security protocols also known as cryptographic protocols are communication protocols which are designed to provide security assurances of various kind, using cryptographic mechanisms. Classic examples of assurances include confidentiality, authentication, message integrity, and most recently anonymity assurances.

### 2.2 Public Key Cryptography

Public key cryptography or asymmetric algorithm is a form of cryptographic keys that usually are designated as public and private key, which are related mathematically. This allows two users to communicate with each other securely and without having any prior knowledge or access to shared secret keys. In the public key algorithm, the private key is kept secret where as the public key may be kept in the open without risk of security exposure. A Famous example that is widely used to describe cryptography is known as Bob and Alice. The example precedes as follow, if Alice needs to send Bob a message. Alice begins by encrypting the data using Bob’s public key, which will turn the plaintext message into ciphertext. When Bob receives the encrypted message, he starts to decrypt the ciphertext using his private key (not known to Alice) and retrieves the secured data. This can be applied to verify authenticity of a message. If Alice wants to encrypt a
plaintext message, she can encrypts it with her private key (not known to Bob) and transmit the message. If Bob is able to decrypt the message using Alice’s public key, he will know the person who sent that message is indeed Alice and she’s the only one who can send it.

To help solve the key distribution and scalability problems, it is helpful to use asymmetric key systems associated with symmetric systems. Usually asymmetric keys are known to be with a wider range of security services than symmetric systems. They help protect against attacks such as interruption, interception, modification, fabrication, confidentiality, authenticity and non-repudiation. However, more computer resources are needed for encrypting and decrypting using asymmetric keys. The speed limitation of WSN presents a significant barrier in using asymmetric systems. Another widely used public/private key algorithm are Rivest-Shamir-Adelman (RSA) and Diffe-Hellman [13].

2.2.1 RSA

Ronald L. Rivest, Adi Shamir, and Leonard M. Adleman developed the RSA algorithm. RSA method depends on the derivation of key-pairs from large sets of numbers, "$N$". "$N$" is the product of two prime numbers, 100 or more digits long. Although used for encrypting entire messages, RSA is much less efficient than symmetric key algorithms such as DES.

To compute a simplified description of RSA encryption, [13] if $N = pq$ to be considered of a two equally sized large prime numbers (n=2 bits each). Where N is set to a size of 1024 bits and $a$ can be factored to a 512 bits. If we consider two integers e,d
satisfying \( ed = 1 \mod \varphi(N) \) in which the multiplicative order is \( \varphi(N) = (p - 1)(q - 1) \) to the group \( \mathbb{Z}_N^* \). The modulus of the RSA can be called \( N \), and \( e,d \) are the encryption decryption exponent. This pair \( \langle N,e \rangle \) would be considered as the public key and is used to encrypt messages. On the other hand the pair \( \langle N,d \rangle \) would be considered as secret or private key and to be known only to the person who is receiving the message. The secret key enables decryption of ciphertext. A message is an integer \( M \in \mathbb{Z}_N^* \). So when the message needs to be encrypted \( M \), the person who encrypts would computes \( C = M^e \mod N \). Therefore, to decrypt the message, the recipient of the message would computes \( C^d \mod N \). \( C^d = M^{ed} = M \mod N \).

### 2.2.2 Diffie-Hellman Key Agreement Protocol

Diffie-Hellman is a protocol or an agreement key protocol which in 1976 two people known as Diffie and Hellman developed this protocol to allows the exchange of messages between any two people without the proper knowledge of each others secret key over an unsecured medium [14].

This protocol it suggests two important parameters known as \( p \) and \( g \), which are both public keys and other nodes in the system can use it. For the two parameters \( p \) is considered to be a prime number and \( g \) is an integer that is less than \( p \), with a property that is known for every number \( n \) between 1 and \( p-1 \) such that \( n = g^k \mod p \). Remember the example of Alice and Bob: if in a situation using the Diffe-Hellman where Alice and Bob need to agree on a shared secret key. It begins by, Alice generating a random integer
private key “a”, Bob then generates a random integer private key “b”. It is then to be followed by the deriving of a public key using parameters $p$ and $g$ and their individual private keys. The public key that Alice uses is considered as $g^a \mod p$ whereas Bob's public key is $g^b \mod p$. The two will complete the set up by exchanging of ones another public keys. The last step would be for Alice to compute $g^{ab} = (g^b)^a \mod p$; and Bob to compute $g^{ba} = (g^a)^b \mod p$ and since $g^{ab} = g^{ba} = k$, therefore, a shred secret key $k$ between Alice and Bob have been developed [14].

One of the concerns that raise an issue with the Diffie-Hellaman protocol is the man in the middle attack. For example, let say Cathy intercepts the public transmission key for Alice, and replaces it with her own public key and sends it to Bob. Then Bob transmits his public key, Cathy intercepts the transmission and replaces it with her own public key and forwards it to Alice. The issue here is that Cathy has agreed on different shared keys with both bob and Alice without the two knowing of this matter. Finally Cathy will be able to decrypt any messages that are sent out by Alice or Bob.

2.3 Secret Key Cryptography

A second cryptography protocol that is widely used is known as a secret key or as a symmetric key. With this key both of the sender and the receiver of the message agree to use the same secret key to decrypt and encrypt. First, when the sender sends a massage it encrypts the message using a secret key, so when the receiver decrypts the message it uses the same secret key. The main challenge of secret key cryptography is the agreeing on a key between the sender and receiver without compromising the key
Table 1. DES, 3DES, AES

<table>
<thead>
<tr>
<th></th>
<th>Key Length (bits)</th>
<th>Plaintext (bits)</th>
<th>Cipher Text (bits)</th>
<th>Strong for use?</th>
</tr>
</thead>
<tbody>
<tr>
<td>DES</td>
<td>56</td>
<td>63</td>
<td>64</td>
<td>No longer</td>
</tr>
<tr>
<td>3DES</td>
<td>112 (2x 56)</td>
<td>64</td>
<td>64</td>
<td>Yes</td>
</tr>
<tr>
<td>AES</td>
<td>128</td>
<td>128</td>
<td>128</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td>128</td>
<td>256</td>
<td>128</td>
<td>Yes</td>
</tr>
</tbody>
</table>

itself. If the sender and receiver are in different geographic locations, a trusted third party needs to be considered such as, a phone system, a courier or some other transmission medium for the initial transmission of the secret key to each other. If an entity manages to intercepts the key in transit, they could have the ability to read, change, or forge the messages that are encrypted using that shared private key. Generating and transmitting keys is known as key management. All cryptosystems must deal with key management issues because all keys must remain secret [14].

Categorizing symmetric key cryptography into being either stream or block ciphers. We begin with stream cipher which it outputs one bit of information at the state of transition. On the other hand, stream ciphers is handled by generating a keystream, that contains a sequence of bits used as a key. When applying encryption it can be done through the combination of plaintext and keystream, also can be defined as the operation of bitwise XOR. To generate a keystream, it can be achieved by either an independent
plaintext state, which yields the ciphertext as synchronous stream cipher, or on the other hand it can be a dependent on the encrypted data.

The second category of a symmetric key is known as a block cipher which is an encryption algorithm that takes a block of readable plaintext and encrypts it into an unreadable ciphertext data in which both texts have of similar length of block, by using the user-provided secret key. Using the same secret key that encrypted the data and using it in the same manner of encryption except reversing the idea can perform decryption in block cipher. The fixed length is known as the block size, which can be 64 or 128 bits and will depend on the processor’s capability.

2.4 Summary

In this chapter, we examined the basic processes of encryption, cryptanalysis, and the different characteristics. Encryption algorithms alone are not the answer to everyone’s encryption needs. Although encryption implements protected communication channels, it can also be used for other duties as well. In fact, combining symmetric and asymmetric encryption often capitalize on the best features of each. Usually when the algorithms are employed as a symmetric key cryptography they can be considered to be faster than the algorithms that are usually employed as a public key cryptography, therefore they are considered to be more of an interest to the purpose of encryption than others. One of the advantages that expensive public key cryptography may have is that the sharing of a secret between entities is not required. In the following chapters,
different security issues and requirements are discussed and we demonstrate the importance of using cryptography in WSN nodes.
CHAPTER 3

WIRELESS SENSOR NETWORK OVERVIEW

In this chapter we will provide an overview to WSN’s network and architecture such as cluster head/frame relay, followed by the obstacles and threats against WSN and the types of security requirements that must be considered. We will conclude this chapter by taking a close look at group key management.

3.1 Introduction

Sensor network is a distributed wireless sensor ad hoc network that contains hundreds or thousands of nodes that are dynamically while using stationary hardware. They are designed for monitoring tasks such as battlefield surveillance, equipment supervision, intruder detection, and wildlife observation, among many others [16], [17], [18].

Typically, the components of a sensor node are considered to have constraints or limitations, whether the fact it have a small storage or memory, not a fast enough processor or the widely discussed issue of power or energy. Different approaches and proposals have been focusing on those constraints to provide a safer and easier approach for the nodes to communicate over wireless a radio link network interface. An important component of a wireless sensor network architecture is the use of a base station, which it is designated as a gateway to process data or important messages from the nodes to the command nodes or data sinks. Neighboring sensor nodes have the capability and ability
to communicate and transmit their readings among them self’s or directly to the basestation if the available security resources are available such as pair-wise keys.

### 3.2 Sensor Network Architecture

Typically, sensor nodes are battery powered with limited memory and a processor capacity. The sensors use radio links to establish communication between the nodes. Other network constraints include a limited pre-configured capability, unreliable communication, limited data rate, limited channel error rate, frequent route changes, unknown recipients, latency or multi hop routing, and population density. They have sensing capabilities: seismic, acoustic, magnetic, and infrared [4]. After the nodes have gathered different readings, they first process and coordinated the data among themselves and then forwarded it to a data sink wirelessly, via other nodes, using a multi-hop path. Once the nodes are deployed they may not be retrievable, therefore, WSN’s life expectancy will largely depend on the life or energy of the batteries. Table 2 [11] shows the hardware of crossbow mica2 sensor nodes.

**Table 2. Hardware of Mica2 Sensor Node**

<table>
<thead>
<tr>
<th>Hardware</th>
<th>Specification</th>
</tr>
</thead>
<tbody>
<tr>
<td>Processor Speed</td>
<td>7.3728Mhz</td>
</tr>
<tr>
<td>Program Memory (Flash)</td>
<td>128Kb</td>
</tr>
<tr>
<td>Variable Memory (RAM)</td>
<td>4Kb</td>
</tr>
<tr>
<td>On chip storage (EPROM)</td>
<td>4Kb</td>
</tr>
<tr>
<td>Off chip storage (Flash)</td>
<td>4Mb</td>
</tr>
<tr>
<td>Digital IO</td>
<td>48</td>
</tr>
<tr>
<td>Analog to digital converters</td>
<td>8 10 bit</td>
</tr>
<tr>
<td>Radio Frequency communications rate</td>
<td>38.4kbit/sec</td>
</tr>
<tr>
<td>Radio power requirements</td>
<td>16mA transmit 9mA receive</td>
</tr>
<tr>
<td>Operating system</td>
<td>TinyOS</td>
</tr>
</tbody>
</table>
As shown in the architecture of a sensor network figure 1 [11], a sensor network consists of a cluster head and a sensor relay node to help organize communication between different nodes in the network.

**Cluster head:** is a sensor node that is responsible for sending data to the base station. Data is gathered through communication with other clusters as well as its own group members as shown in figure 1. Cluster heads are responsible for aggregating data, assigning identification to each sensor node in the cluster, accommodating newly joined nodes, and taking care of the failed nodes.

**Relay sensor nodes:** are positioned between different clusters and are primarily responsible for passing information from one cluster head to another. By minimizing resource consumption, sensor networks can become more reliable. They can be self-organized with centralized or decentralized controls and become more robust and able to recover from potential failures.

One major concern with regards to WSN is its energy inefficiency. We already discussed the importance of a node’s battery life. The problem is that under many circumstances (e.g. battlefield surveillance or sensors dropped from an aircraft), it is not possible to recharge or swap the drained batteries. Thus, energy efficiency is a hot topic of discussion and explored in many research papers all over the world, the most important thing remains, is how can we conserve the energy or power of anode in order to expand the life of it.
A sensor node’s radio can be in one of the following four states: transmit, receive, idle, or sleep [19]. Usually a node is considered to be idle if the transceiver is neither transmitting nor receiving data and is said to be sleeping if the radio is completely turned off. A study that was performed by WINS [19], it considers that rockwell seismic sensors indicate the consumption of power for the transmit state is between 0.38W and 0.7W, and for the receive state is 0.36W. The idle state is 0.34W and for the sleep state 0.03W. In other words, whether the node is receiving data or is idle, it will consume the same level of energy as when it is transmitting data. Different power saving techniques are performed, one of those techniques is known as the sleeping mode scheduling, which nodes can switch between sleeping and active mode and the adjusting of the transmission range of the nodes in order to gain power control, routing and data gathering to reduce the sum of transmitted data.
Using the sleeping mode, nodes consider the technique of time division multiple access (TDMA) since a node can power down or sleep between its assigned time slots, this technique is widely used for power conservation. Generally, TDMA is known to be a digitally transmitted technology through the allocation of unique time slots within the channels allowing the access of a single radio-frequency (RF) channel that doesn’t contain any interruptions between the users. The TDMA digital transmission scheme multiplexes three signals over a single channel.

A hot topic of research is in the area of active power control, where the idea of the node selecting its own individual transmission power along the cooperation of other nodes. The problem that can be generated from a feedback control is known as a decentralized problem. An important issue that nodes encounter in the even of too much power is congestion, however if a larger transmission range is used by each of the nodes this problem can be avoided and the network will remain connected. When \( n \) nodes are considered to be randomly distributed in a disk, the network is assumed to be connected. The range of the transmission \( r \) of all nodes is selected using

\[
 r \geq \sqrt{\frac{\log n + \gamma(n)}{\pi n}}
\]

where \( n \gamma(n) \) is a function that approaches infinity as \( n \) becomes large.

WSN lack the physical infrastructure and have a dynamic topology. Most of the existing work on WSN security protocols are not energy efficient for networks that change dynamically or require a quick re-establishment of group keys, as the limitation of
resource may interfere with a secure communication. WSN architectures are organized in hierarchical and distributed structures as shown in Figure 2 [2].

3.2.1 Hierarchical Sensor Networks

In hierarchical sensor network there is a hierarchy among the nodes based on their capabilities. Basestations are usually considered to be more powerful than regular sensors or cluster heads in a network to the fact they are a powerful data processing and storage center for an access point for a human interface. It is considered to be the gateway between networks or clusters, they collect sensor readings, perform costly operations on behalf of sensor nodes and manage the network itself. In some applications, base stations are assumed to be trusted and temper resistant. Thus, they are used as key distribution centers. Sensor nodes are deployed one or more hops away from the base stations. They form a dense network where a cluster of sensors lying near each other may be providing similar readings. Nodes with better resources, also known as cluster heads, may be used for collecting data, managing local traffic, and sending information to base stations. Transmission power of a base station is usually enough to reach all sensor nodes, but sensor nodes depend on the ad hoc communication to reach base stations; Hierarchical Wireless Sensor Networks(HWSN) as shown in Figure 2 (a) can be considered vulnerable to attacks such as spoofing, jamming, and replaying.

3.2.2 Distributed Wireless Sensor Networks

In distributed sensor network, there is no permanent infrastructure and the topology of the network usually not known to anyone prior to the deployment of nodes.
Figure 2. WSN Architecture

Sensor nodes are usually randomly scattered all over the target area, and once deployed, each sensor node scans its radio coverage area to map its surrounding neighbors; as shown in Distributed Wireless Sensor Networks (DWSN) of Figure 2 (b).

3.3 Wireless Sensor Network Security

WSN security is one of the most important topics in securing the communication between the sensors. Because of widely used networks intrusions and attacks have become threats. Recall that WSN transmits through air on a non-secure Radio Frequency (RF) channel. Attacks on a wireless network can be categorized as active or passive. Active attacks involve some modification of the data stream itself or creation of a false stream. Passive attacks occur mostly in form of eavesdropping on transmissions. Although expecting the deployment of a 100% intrusion-free network may be unrealistic (expensive), network security can be improved through less expensive means where the security of the entire network is maintained but the local attacks are ignored.
The main requirements for developing security protocols in WSN are: authenticity, availability, confidentiality, freshness, scalability, integrity, flexibility, and self-organization. In order to maintain confidentiality, messages and keys must be protected. To verify authenticity, the sending and receiving nodes must be able to encrypt and decrypt messages without disclosing the contents to unauthorized nodes. Refreshing the keys is another critical function performed in a network. In order to maintain network integrity, the unauthorized nodes should not be able to participate in computation of these shared keys. The security protocol must be scalable and have the ability to handle large networks, keys must be readily available to authorized nodes, and keying protocol should work autonomously once the network has become organized. It must be flexible with the applications.

3.3.1 Obstacles of WSN security

A WSN contains different constrains and obstacles that can be challenging at times to the administrator. To develop useful security mechanisms, we need to recognize and understand these constraints and their issues [23]:

- **Memory and Storage Limitation:** Sensors are considered to be limited with memory and space. Therefore, we need a small size algorithm in order to build security mechanism since the processor is not fast enough as well..

- **Energy Limitation:** This is a very important aspect of sensor nodes and at times it is considered to be the most important thing in deploying nodes. The assumption that a node is irretrievable after it is deployed is made; therefore they
are not easily replaced or recharged, which makes conserving the energy in the node is important to expand its life.

- **Unreliable Transfer:** Since the communication between the nodes is considered to be radio linked or wireless, the data that is being exchanged between the nodes may be lost or compromised for reasons such as jamming, spoofing, or simply dropped.

- **Latency:** The dynamically changing of nodes once they are deployed and the frequent changing in routes or hops can lead to the latency of the network, this can cause the difficulty issue of synchronization between the nodes. It is important to synchronize between nodes to achieve a secure key distribution and event reporting.

### 3.3.2 Security Requirements

Security is an important issue and aspect in today’s life whether we are using a wireless or wired network. You have thousands of important information or data that is being transmitted over networks on daily basis, and the chances of that data being compromised is always high do to the fact of not sufficient security on the network. Therefore, we consider the following requirements as important steps to consider before trying achieving an excellent security for the network. The following are few of the security requirements that are considered to be important to achieve security protocol in WSN [23]:

22
- **Data confidentiality:** Usually this is the first problem that is addressed in security since it is considered to be the most important issue. Depending on the situation the node might be in, the data it's holding could be highly sensitive data such as in a battlefield, where the nodes do not distribute any data between its neighboring nodes, therefore the information that is not disclosed to others considered to be confidential.

- **Data Integrity:** Protecting the data is probably the most important issue after confidentiality. Since an entity could try to tamper with data through compromising a node, in which in this case data can be lost, changed, or maybe just disclosed to an unknown entity. Such cases may occur in a hostile communication environments or a non-hostile environment, therefore, the integrity of the data ensures the safe protection of the data without being tampered with.

- **Data Freshness:** This requirement suggests the freshness of a message by constantly updating the transmission between nodes, in which it specifies old messages to be transmitted or replayed. Data freshness is needed when shared-key strategies employed in the design. When a sensor is unaware of a new key change, the normal operation of the nodes can be tampered with. Solving this problem, the implementation of a time-counter into the packet can guarantee data freshness.
**Authentication:** It specifies the true identity of which the information was received from. Basically an attacker may pretend to be someone they are not and transmit false sensitive information, or perhaps try changing the coarse of the stream by injecting additional packets. Therefore, the receiver must be able to determine if the data received is indeed from the correct source.

**Availability:** Since sensor nodes are considered to be a low cost solution, implementing or adjusting encryption algorithms to accommodate the wireless sensor networks can be a costly matter. Therefore, the reuse of software code may be considered to some but not all through simple modifications to the code. Whereas, others may require the establishment of additional communication for achieving its main objective.

### 3.3.3 WSN Attacks

Sensor networks are particularly vulnerable to certain types of attacks [20]. Different types of attacks can be performed on a WSN such attacks may include node compromising, multihop attacks, denial of service by jamming the network etc., these attacks can happen do to the fact of poor physical protection. In this section, we will address some common denial of service problems and move on to describing attacks on the routing protocols, identity based attack known as the Sybil attack, and more [21, 22, 23, 24]:

- **Denial of Service attacks:** This type of attack is known for jamming radio signals between the communicating nodes, which interferes with the transmission
of data. This attack can be classified into two things constant jamming, and intermittent jamming. In the event the entire network is jammed without the ability to control it or transmitting any messages would be considered as constant jamming, whereas, if the nodes are periodically able to transmit messages it would be considered as intermittent jamming.

- **The Sybil attack:** This type of an attack is described to be a malicious device with the ability of inheriting multiple identities. This attack was described originally to have the ability to defeat the redundancy mechanisms between data storages networks such as in a peer-to-peer environment. Additionally the sybil attack is considered to be effective against data aggregation, routing algorithms, voting, and the detection of misbehavior.

- **Traffic Analysis Attacks:** In WSN the gathering of data by a single node is not considered to be unusual. Typically sensor nodes are designated as many low-power sensors that have the ability to communicate to a robust and powerful base stations. This type of communication is considered to be normal unless the attacker has the ability to disable the base station. Deng et al. have described two different attacks that can identify the base station and lead to its capturing or compromising in the network without needing to understand the contents of the packets.

- **Privacy Sensor Attacks:** While the different promises to the increase in automatic data collection is considered, they tend to pose significant potential for
abuse. In particular the relevant concerns of the nodes tend to be a privacy problem, since the sensors can perform the increasing of collecting data.

3.4 Defending against attacks

To satisfy security requirements and prevent WSN from malicious attacks, different preventive measures should be taken to defend and secure the nodes. We will now consider key-establishment defense in WSN, which is at the foundation of its security and based on asymmetric key algorithm. We continue our discussion by defense against Denial of service (DoS), secure broadcasting and multicasting, attacks on data aggregation, routing protocols, trust management intrusion detection, sensor privacy, and physical attacks.

Pre-deploying keying is still considered today the most useful and dependable approach for bootstrapping. Generally, the nodes are pre-configured before they are deployed and loaded with any keying for its protection. Different pre-deployment proposals have been introduced, some of those approaches include: global key shared by all nodes [25, 30], sharing of a unique key with the base station [26], and random key sharing [29, 27, 28]. From the security point of view an ideal approach would be the establishment of pair-wise keys that can be shared between two neighboring nodes to achieve a secure communication. Using this method provides security to the network by not revealing the different keys that other nodes may have in the event of compromising of nodes. However, this approach requires every other node in the network to have a unique key identifying to communicate with it. In WSN, neighboring nodes are not
known in advance, thus these pair-wise shared keys will need to be established after the network is deployed.

3.5 Group Key Establishment Protocols

Classifying group key establishment can fall in to two categories the first is known as key management (key distribution) and the second as key agreement. In key management protocols, creating and updating group keys can be achieved through a central key server. Therefore, securing the distribution of keys among nodes. Basically each node will hold a partial of that group key that has been distributed among the nodes whereas the updating and the creating of the keys will be handled through the joint responsibility of the members.

3.5.1 Group Key Management

Establishing and maintaining of the group keys falls under the umbrella of key management. Maintenance activities consist of changing the group keys whenever a new member is introduced or an existing member is expelled [33]. To deploy security services a policy that can produce a good key management is consider. In such cases to generate group keys are either centralized or distributive. In the event of centralized, the group key is generated and distributed by an entity, whereas the distributive all the member contribute to generate the key. However the distributives can be fault tolerant and therefore, diminishes the risks of vicious key generation by a single entity.

3.5.2 Tree-Based Key Management Protocols
In key management protocols, a central key management server is usually present in the system, and group keys are initialized, updated and distributed by the key management server. Wong et.al, Proposed a Tree-Based key management protocol, as an example of key management protocol. This protocol assumes a multicast group of \( n \) members, \( M1 \) through \( Mn \), and there is a central key server in the system that manages membership and key assignment. The central key server stores a key tree, as shown in Figure 3 [32]. Each node also stores keys from the leaf up to the root in the tree, for example, in Figure 3.2, \( M9 \) stores \( k9, K_c \) and \( K_{root} \). In the event of a new node joining the group, the key server adds the new member as a leaf node in the key tree, updates all the keys from the new node up to the \( K_{root} \) in the key tree and securely distribute them, for example, in Figure 3.2, if \( M9 \) is the new member, then \( K_c \) and \( K_{root} \) in the key tree should be updated. Updated \( K_c \) will be encrypted with \( K_{c\_Old} \) and multicast to \( M7 \) and \( M8 \) (\( (K_c)K_{c\_Old} \Rightarrow \{M7, M8\} \)). Updated \( K_{root} \) will be encrypted with \( K_{root\_old} \) and multicast to the whole group (\( (K_{root})K_{root\_old} \Rightarrow G \)). Updated \( K_{root} \) and \( K_c \) will be grouped together, encrypted with \( K9 \) and unicast to \( M9 \) (\( (K_{root}, K_c)K9 \Rightarrow M9 \)). Key update at node leaving is more complicated. All the keys possessed by the leaving node should be updated in order to secure future communication. For example, in Figure 3.2 if \( M9 \) is the leaving node, then \( K_c \) and \( K_{root} \) should be updated because they are held by
Figure 3 Tree-Key Based Management Protocol

$K_{\text{root}}$.

$K_A$ and $K_B$ are not affected by the leaving node and can be used to encrypt the updated key(s). Updated $K_{\text{root}}$ is encrypted with $K_A$ for $M1$, $M2$, $M3$ ($K_{\text{root}} \rightarrow \{M1, M2, M3\}$), and encrypted with $K_B$ for $M4$, $M5$, $M6$ ($K_{\text{root}} \rightarrow \{M4, M5, M6\}$). Updated $K_{\text{root}}$ and $K_C$ are grouped together, encrypted with $K7$ for $M7$, and encrypted $K8$ for $M8$ ($K_{\text{root}}, K_C \rightarrow K_7; (K_{\text{root}}, K_C) \rightarrow M_8$) [32].

### 3.6 Related Work

Researchers and other experts who are interested in WSN technology and its varied applications have presented numerous proposals and strategies for optimal group key management for WSNs in recent years. In general, the group key management strategies that have been proposed and implemented to date can be assigned to one of two
categories: centralized or decentralized. Group key management protocol, One-way Function Chain Tree (OFCT), and Efficient Large Group ELK are all centralized strategies, while MARKS and Kronos are decentralized strategies. What the centralized and decentralized strategies have in common is their goal of attempting to strike an acceptable balance between performance of the WSN and its particular demands and needs, particularly with respect to storage capacity. At the same time, the centralized and decentralized group key management strategies tend to be limited by two common shortcomings as well. Both types of strategies generally fail to include a mechanism for compensating for devices or nodes within the system that are either damaged or which are malfunctioning, whether through the wear and tear of natural use or as the result of malevolent interference. Both types of group key management strategies also generally fail to consider the cost—both financial and energy-wise--of support key management requirements, which include initial setup and periodic re-keying.

One notable exception to these claims is the Hybrid Key Tree (HKT), a strategy that attempted to manage achieve security and efficiency simultaneously by relying upon a dual-level hybrid key tree. The strength of the HKT strategy is that it endows the WSN system administrator with an impressive degree of control due to the feature of the HKT’s sub-linear storage complexity, achieved by the use of clusters. Specifically, the system administrator is able to reduce threats to the WSN by adjusting cluster sizes as needed. Despite this impressive advancement, the HKT strategy is not without its own limitations. In particular, the HKT strategy does not have a mechanism for detecting or
preventing collusion and other threats among devices that are active. However, describing how to establish a pairwise key was not described therefore, the new strategy that is proposed herein is unique in this regard because it decreases the capturing of the devices and nodes in the WSN, thereby minimizing threats.

Some approaches to improve the rekeying process were proposed by Younis. As many of the strategies that have been proposed for rekeying between nodes are based in part upon the Exclusion Basis System (EBS), which was first posited by Eltoweissy and colleagues. Limitations of EBS as it was originally conceptualized is that it was unacceptably open to collusion-type attacks. With Younis proposed known as SHELL addressed this problem of the EBS and how to protect it from collusion attack. However, serious notation that Younis did not describe in their paper was the establishment of pairwise key between neighboring nodes in their system initialization procedure, which can’t make their applied approach to be fully secure.

Re-keying is a common problem. Jolly et al’s key management proposal, while notable for its low-energy requirements in terms of key set-up and management, is terribly inefficient with respect to re-keying. The high number of messages that are required prior to assigning a new key or renewing an existing one undermines the efficiency and efficacy of their system. Jolly et al’s model also centralizes key management power in a single command device. Recognizing the limitation of this model, the model proposed by this dissertation is sensitive to the fact that a single node cannot achieve utility or security for WSNs.
There have been many other key generation and distribution strategies that have been proposed and tested for use in WSNs. Eschenauer and Gligor, for example, devised a key pre-distribution strategy based on probability theory. According to their strategy, a ring of $n$ keys, selected at random from a larger pool of keys, would be assigned for the nodes in the network. To have neighboring nodes within the network to communicate with one another, they would be required to match up a common key that could be found on each of their key rings. In the event that the two nodes did not share a common key, a third party would be activated and involved as a broker.

Subsequent researchers and developers have built upon the model proposed by Eschenauer and Gligor. Chan and colleagues suggested three novel strategies of key pre-distribution intended to increase the network’s resiliency with respect to node capture. Similarly, Liu and colleagues explored and manipulated the variable of node location as a core feature of key selection. Du and colleagues offered a key pre-distribution approach in which the probability that un-compromised nodes will be affected is minimal when the system meets a threshold condition for the quantity of nodes that have been compromised. While each strategy has built upon the former ones, the glaring limitation of each is that they fail to make provisions for re-keying, which is a necessary aspect of WSN utility and security.

A review of the existing key management strategies also reveals that there has often been an uncomfortable trade-off between the desire to improve the security of the WSN system and the desire to improve the system’s efficiency and reduce its resource
demand burden. Park and colleagues, for instance, made such a trade-off in their model, which was a lightweight security protocol for WSNs. Their model is efficient, relying upon a hierarchical keying system; however, the reliance upon an external but trusted third-party for authentication may limit its security value. The strategy that will be proposed here rejects the notion that this trade-off must be a feature of key management strategies. It is the researcher’s belief that both security and energy and resource efficiency can be achieved through creative reconsideration and re-strategizing.

3.7 Summary:

A quick summary about the topics we have discussed during this chapter, in which it covered wireless sensor network overview, wireless sensor network security issues, requirements, attacks, and other related works to our scheme. We talked about an overview of different issues or problems that may be encountered such as energy/power, security, attacks etc. and in order to achieve a top security we have to work along side with the current resources that are available. Different solutions have been proposed for security, therefore, in the following chapters we will present our scheme and their role to the WSNs.
CHAPTER 4
PAIR-WISE KEY ESTABLISHMENT

In this chapter, we present our pair-wise key scheme, with the objective of enhancing network survivability against node capture with the least amount computation, communication, and storage overhead. We assume that the command node is secure but the gateways and sensors can be compromised by an adversary. We will also give our assumptions on capabilities, describe the role of each node in the security framework, and give details of network initialization and establishment.

4.1 Threat Model

WSN system management strategy must consider the possibility that network devices, hereafter referred to as nodes, can be captured and compromised by a hostile entity -- internal or external to the system -- with the ability to manipulate the system and its components. One cannot make the assumption that the WSN or any single device within it is secure even though system administrators always strive to minimize risk. The threat model assumes that there are various threats to the WSN system, sensors or nodes can be captured, appropriated, and manipulated, their keys can also be identified thereby threatening the integrity and security of the entire system (four of the security concerns were discussed in Chapter One).

There are multiple threats to the safety and security of a network including but not limited to divulgation of keys through the act of spoofing, degradation of the gateways
processing and inter-node transmitting capacities, and appropriation and misuse or abuse of the network. Therefore, our main goal is to create a safe and secure environment for neighboring nodes (or nodes in close proximity) to communicate and exchange data. Younis et al. [42] attempted to address this security issue without the establishment of pair-wise keys. The gateways’ linking nodes may be the most vulnerable aspects and addressing it is very important to the system despite the fact that they are more difficult to penetrate than a single node, but the establishing of pair-wise key between neighboring nodes in this process is as important, do to the fact that it demonstrate an equal level of security and the reduction of node compromising and the safe of exchanging data among them.

Recall our assumptions that the hostile force has a limited but compelling potential to compromise the entire WSN system. Our model further assumes that there is no way of identifying in advance, exactly what data each single node warehouses; therefore, the adversary cannot target nodes selectively for content. Another assumption of the proposed model is that the hostile force aims to capture the system’s keys in order to reveal or gain knowledge about the entire operation of the WSN. Thus, our model proposes the establishment of pair-wise key in order to prevent a hostile force from compromising the nodes individually or take advantage of collusion among the degraded nodes which may lead to the capturing of the network. There are a number of different ways that such nodal corruption can occur. One of the central notions of the proposed strategy is that thoughtful pairwise key establishment assignment will reduce the
likelihood that any node can be compromised. Although security threats and weaknesses will always be likely to exist in WSN management, the present strategy focuses on pairwise key assignment that reduces network degradation.

4.2 Network Model

Existing models of random key pre-deployment in wireless sensor networks generally assume that each constituent node within the network can be in dynamic communication and interaction with its neighbors. Visually, such a network is presented in either a grid format or a graph format in which the nodes are distributed randomly. It is also assumed that inter-nodal communication can only be constrained by key-matching. Such an assumption, however, is naïve and likely to limit the utility of the WSN. Limitations of current model random key-deployment models include:

- Current assumptions do not account for variables such as barriers that block signals during transmission from one node to another. In an actual WSN application, however, there are many types of communication-blockers such as physical structures (walls and buildings) or topographical features (mountains and hills) that can disrupt a system and reduce its efficacy.
- The ability of system administrators to determine or pre-arrange a node’s position with respect to another node is limited at best, due to distribution methods.
- Nodes can be added to the WSN at any time after the system itself has been deployed, thus confounding the original nodal arrangement and inter-nodal relationships.
• The nodes in a network tend to be homogenous in their capability, thus, one sees that the assignment of specific keys for specific neighbors is simply not a feasible strategy.

This paper will briefly explore the importance of the proposed model by Younis et al. [42] and how it ties in with our model. Most of the strategies that have been proposed for rekeying between nodes are based in part on the Exclusion Basis System (EBS), which was first introduced by Eltoweissy et al. [40]. EBS is a key management approach that is combinatorial in nature and as such, it is ideal for the management of multi-nodal networks. EBS relies upon the advantages offered by the quantity of administrative keys, “k”. However, one of serious limitation of EBS as it was originally conceptualized is that it was unacceptably open to collusion-type attacks. Younis et al. [42] proposed an approach or schema known as SHELL which addressed this problem and offered a solution to protect it from collusion attacks. However, as mentioned earlier, Younis did not describe in the establishment of pair-wise key between neighboring nodes during the system initialization procedure, also known as network bootstrapping (see section 4.4). Our model explores establishment of a pair-wise key that addresses neighboring nodes. In doing so, we will reference a paper by Zhu [49] that demonstrated the feasibility of this approach for small as well as very large systems, which works well with Younis’s dependence on a network in its entirety.

The proposed model consists of a network that constituent nodes that represent a range of potential applications and functions. There are two basic categories of nodes
that comprise the system: sensing nodes, which will be referred to hereafter as N1 nodes, and memory and processing nodes, which will be referred to henceforth as N2 nodes. The functions of these two nodal groups are different and the fact that each is dedicated to a specific type of task maximizes the performance of all nodes. The purpose of the N1 nodes is to collect data. As its name suggests, the sensing nodes (N1) perform the sensing function in the network system. In contrast, N2 nodes have memory and processing capabilities that the N1 nodes lack. The N2 nodes are programmed with more keys than the N1 nodes and this increased functionality and task-specificity permit the N2 nodes to serve as gateways that link networks.

N2 node devices are outfitted with “armor” that limits tampering by potentially hostile forces. The N2 nodes also have a rapid encryption and deletion algorithm, the purpose of which is to protect keys from misuse if they are compromised by a hostile force. All of the nodes in the network are able to communicate with neighbor nodes when they share a communication key in common. In some networks, however, the dispersion of nodes may be more expansive than the transmission reach of a single node. Regardless, inter-nodal communication cannot occur unless the N2 node functions as a gateway that facilitates the transmission process. These gateway nodes are able to connect with backhaul, which provides a foundation for a skeletal network.

4.3 Network Architecture

A typical WSN consists of different components, each of which has their own functionality. In this section we are describing the components and functionality of the
architecture of the network in order to establish pair-wise key. We will discuss three
types of components: command node, gateways, and the typical sensor nodes [40].

4.3.1 Command Node

The command node is responsible for task delegation for sensors. One of the tasks
of sensors is to gather data and transmit it to a gateway, which in turn processes and
transmits the aggregate data of the sensors to the command node. There are two
important assumptions that are made about the command node: (1) it is a trusted entity
and (2) unlike nodes and gateways, it cannot be compromised.

Like the gateway, the command node has a specific set of responsibilities within
the network. They are as follow:

• Serve as an archive in which all of the valid IDs and pre-loaded keys (such as
  $K_{sg}$, $KS_{CH}$, and $KS_{Key}$) of the sensors are collected and maintained.

• Authenticate both the gateways and their individual sensors. This responsibility is
two-pronged. In other words, the command node must authenticate gateways and
sensors when a WSN is initiated, as well as when it is changed or modified in any
way.

• Generate and assign keys that facilitate inter-gateway communication and that
  also detect the compromise or failure of any constituent system gateways.

• Perform periodic key renewal for inter-gateway communication, as well as initiate
  the renewal of gateway-to-sensor communication keys. This latter function
  combats the threat of spoofing.
Individual sensors comprise a cluster, which in turn are managed by a gateway.

4.3.2 Gateways

Individual gateways have the power to engage in communicative transmissions with other network gateways when they are linked by shared keys. These common keys are:

- $K_{sg}$: This designates the sensor discovery key. Every gateway has knowledge of the sensor discovery key.
- $K_{gy}$: This designates the pre-loaded key of the individual gateway. This key facilitates direct communication between the gateway and the command node.
- $K_{gi;gj}$: This designates the gateway key that permits communication between distinct gateways, indicated here as gateway $i$ and gateway $j$. The command node provides the gateway with this key during the bootstrapping phase.

Every gateway functions as the head of a cluster of sensors, each of which are assigned unique communication keys that permit the encryption of data for message-sharing. The fact that the constituent sensors of a cluster share communication keys is important because it promotes energy-efficiency. Furthermore, messages can be encrypted and decrypted, thereby facilitating the process of aggregation. Nodes of a cluster can also be assigned to sub-groups, each of which will carry their own unique set of communication keys that facilitate unique sub-group functions.
Figure 4 Network Architecture

Each gateway performs specific security functions to protect the integrity of the nodes in
the cluster and their keys. These functions are as follow:

- Generate administrative keys for external clusters as needed.
- Refresh data keys of the cluster after network setup.
- Detect and eject any sensor nodes in the cluster that has been compromised.

4.3.3 Sensors

The gateway has the power to communicate with each node in its respective
cluster. The elements of the cluster in turn communicate with each other by means of
one-hop or multi-hop transmissions. The individual ID of each node is stored within the
command node’s central database where each node has a commonly programmed
discovery key, $K_{sg}$. Each node also has a one-way hash function that can be used to re-
key $K_{sg}$. In addition to these commonly shared features, each node also has two pre-
loaded keys, $KS_{CH}$ and $KS_{Key}$, which are assigned for the purposes of initial key distribution. Finally, all nodes are able to store administrative keys (K) and communication keys (c). The boundaries and limits of the administrative and communication keys can be programmed based on specific needs of the system with respect to bandwidth and memory. The model makes the assumption that the gateway for a cluster has the power to identify any nodes that become compromised and recognizes that the gateway itself can become compromised as well. The model further assumes that the gateway, which is also known as the command node, can conduct an investigation and repair procedure once it identifies a compromise in any single node.

4.4 Network Bootstrapping

The phase of system initialization that precedes actual operations is referred to as network bootstrapping. There are four distinct and critical steps in the network-bootstrapping phase: (i) Sensors are implemented; (ii) Sensors constitute clusters; (iii) Neighboring nodes. Once gateways are registered they begin to communicate with one another. Each of the stages in the network bootstrapping phase is discussed below.

(i) **Sensor Implementation**: Sensors are initiated when they begin to convey their ID and location to their assigned gateway. This communication from the sensor to the gateway is encrypted and uses key $K_{sg}$. The proposed model makes the assumption that the system is equipped with adequate numbers of gateways to provide a hosting platform for each of the constituent sensors. The gateway
receives the sensor’s message, decrypts it, and logs the information. When the sensor implementation and recognition is complete, each of the sensors generates a new $K_{sg}$ through the one-way hashing function, which is known to both the command node and the sensor nodes but not to the gateway. Ultimately, the initial $K_{sg}$ key becomes obsolete as this stage ends.

(ii) Cluster Assignment: During this step, the sensor nodes are assigned to their respective clusters, each of which are administered by a gateway, designated as $G_{CH}[i]$, where $i$ is the cluster number. Clusters are formed by gateway collaboration according to different sets of criteria, which may include the physical or geographical position of a sensor, its communication range, or its type. Whichever the case may be, the sensor must be able to reach its gateway. This step is completed once every gateway has registered the ID and location of each constituent sensor and has put the sensors into contact with one another.

(iii) Gateway Registration: During this phase, each gateway connects with the command node. Just as the sensors did with the gateway itself, the gateway transmits announcements about its ID and its location to the nodes; this information is encrypted using the pre-loaded $K_{gc,i}$ key. As the command node receives each gateway’s identifying information, its sets up the provisions for inter-gateway communication and transmits the inter-gateway keys it will use for this purpose, which are encrypted by $K_{gc,i}$. This process is critical, as the
gateways use their inter-communication keys to constantly transmit information to one another about potential system compromises. Finally, the command node conveys the initial sensor contact information to each gateway using the $K_{sg}$ key.

**(iv) Neighboring nodes:** The final step of network bootstrapping consists of pairwise key establishment models that are proposed in [49] and applied to the model proposed by Younis et al., [42]. The combination of the two models fit perfectly and not only provide an equal level of security but also reduce the consequence of nodes being compromised. One of the assumptions of the models is that any node can be assured of confidence in communicating safely with its neighbors if and when the condition of a shared key exists, thus confirming security.

- **Backhaul** – Proposes that an N1 node can only have confidence in an N2 for data sharing purposes and with respect to setting session keys. In this model, an N1 node will only exchange data or establish a session key with an N2 node if they have a common key that can be matched.

- **Neighboring nodes with Limitation** – Allows two N1 nodes to set a session key and exchange data. N1 nodes in this model also permit communication with N2 nodes that are neighbors. In a situation when an N1 node and an N2 node are not neighbors, their communication will need to be brokered by a trusted third-party,
another N2 node that can set a session key. Thus, one understands why this is a limited trust model.

- **Neighboring Nodes Open Trust** – As the name suggests, the confidence among nodes is high. N1 nodes in the same neighborhood trust one another and can share data. If, for some reason, two N1 nodes cannot establish a session key, another node, either N1 or N2, will be used as a mediator and broker.

The proposed model accounts for scenarios in which there is only one node needed for key establishment or, in contrast, an unlimited number of nodes that can participate in the key establishment process. In either situation, the backbone architecture is considered secure. There is another assumption that is important to this model, and that is that the session keys are assumed to be deleted once the communication process has been fulfilled. This function decreases the vulnerability of key compromise as both the communicating and recipient nodes re-key upon completion of their exchange.

### 4.5 Pair-wise Key Establishment

Younis et al. [42] had proposed an approach or schema known as SHELL which addressed collusion attacks. However, one of the serious short-comings of the SHELL model as proposed by Younis is that it did not explain the establishment of pair-wise key between neighboring nodes in the network bootstrapping. In the previous sections, the network-bootstrapping phase was briefly explained and in this section the methods that are used for establishing shared keys among neighbor nodes will be addressed. The
rationale for establishing a set of protocols for shared key interaction is to improve security and decrease nodal vulnerability. Therefore network assume to pursue normal operations afterwards.

All members of the system are aware of the keys and the process through which they need to communicate with each other or with the system at large. The members of the system will be called upon to deploy this information during the course of normal network operations both when re-keying is required and when new nodes are added to the network by using [42].

4.5.1 Key Distribution

The proposal for the unbalanced distribution of keys among the nodes in a WSN draws upon the theoretical model proposed by Eschenauer et al. [50]. This model makes provisions for the following formula: Where $P$ represents the total key pool, a key ring of the number of keys in each $N_1$ node is represented as $k$, and a key ring of the number of keys in each $N_2$ node is represented as $m$. The following equation is based on probabilities and can be found in [48]:

\[
P[\text{Match}] = 1 - \frac{(P - K)!(P - m)!}{P!(P - m - k)!}
\]

4.5.2 Network Connection

In this section, we will discuss the rationale for establishing a set of protocols for shared key interaction to improve security and decrease nodal vulnerability as well as the network connection of our proposed model and the two trust models ([49] and [42] as
referenced in this document). In addition to the contribution is to demonstrate scenarios of how each case is handled:

- **Backhaul** – In order to create the conditions necessary for backhaul, an N1 node and an N2 node must share a common key. There are two possible scenarios to be considered here: (a) There is only one N2 node in the entire neighborhood, (b) There are multiple N2 nodes in the network neighborhood. While the first scenario suggests that a single N2 node will fulfill that function, the second scenario postulates that every N2 node in the neighborhood can function as a gateway. N2 nodes are always considered trustworthy and they contain a higher number of keys than N1 nodes. For these reasons, one can collect theoretical and practical evidences supporting the claim that N2 nodes can communicate with one another with exceptional reliability.

- **Neighboring nodes with Limitation** – This scenario permits inter-nodal communication between two N1 nodes in the same neighborhood. The N1 nodes can only communicate with one another however if they share a common key or can establish a key for that session through an N2 node. The security and reliability of the N2 nodes allows us to use any arbitrary N2 node for this purpose.

- **Neighboring Nodes Open Trust** – To establish connectivity in this scenario, N1 nodes can establish session keys in one of the two ways: (1) by using an N2 node or path of nodes or (2) through other N1 nodes with a shared key. Existing research on this subject has examined the use of up to $n-1$ hops (where $n$ is the
number of nodes in the system) to establish a session key; however, such a long path for session key establishment presents at least two disadvantages. The first disadvantage is intolerably high latency, a threat that is particularly likely in situations where nodes are highly mobile. In such situations, nodes are frequently required to establish new session keys even in mid-session. The second disadvantage of the multi-hop system is that it can introduce an unacceptable number of intermediaries into the session-key establishment process. These intermediaries can compromise the security of the exchange that is occurring by listening in on the session key establishment process. The decryption of privileged communications is one consequence of this model. As the communication chain becomes longer, the probability of existing compromised nodes increases exponentially with every hop.

4.6 Protecting Against Attack and Failure

In this section the evaluation of the impact of a node’s compromise are presented and discussed. It is important to keep the fact that all nodes that are in communication with one another are always required to set a new key for every session in mind. This is also true for situations where the nodes share a key in common. In addition, all nodes that are involved in establishing the session key will delete the key information once the session has been initialized by establishing communication. These checks and balances in the nodal system prevent vulnerable nodes from becoming victims of eavesdropping and key or nodal compromise.
However, all threats are not eliminated from the inter-nodal communication process. It is entirely possible that compromised nodes can re-key and thereby introduce deceptive data into the system. It is also possible that by facilitating the establishment of a key for a session, a node can eavesdrop on the session and compromise the system. The degree of this type of vulnerability is calculated by examining the total number of nodes with relative to the number of compromised nodes that possess the means to establishing session keys. For this reason, it is important to evaluate the capacity of the communicative ability between compromised nodes with healthy neighbor nodes. The following analysis focuses on peer-to-peer with liberal trust scenarios in which there is a single intermediate node and all nodes with the keys that are needed to reach connectivity under usual circumstances are deployed.

4.6.1 Node (N1) Compromise

Almost by definition, WSN will always be deploying sensor nodes in high-intensity, high-risk settings and situations. As noted earlier, this is both the strength and the vulnerability of the WSN system. Because sensor nodes are introduced into environmental settings that are often chaotic and physically, geographically, or climactically vulnerable, system administrators must recognize that nodes are prone to being tampered with or even compromised completely. Because N1 nodes, as noted earlier, perform data collection functions but do not process or convey data to the network administrator, N1 nodes lack the kinds of architectural and structural protection that is characteristics of N2 nodes, which are more resistant to physical interference. For
this reason, it is important to arrive at an understanding of how keys that are revealed when an N1 node is compromised can impact the security of the entire WSN. In a WSN that is operating optimally, nodes are in dynamic interaction with one another, creating session keys with neighbor nodes and eliminating session key information once communication has been established. In an ideal situation, the system is resilient to threat but it is rare for WSN to be used in a situation that is considered ideal.

One way of understanding the impact of nodal compromise is by studying connectivity. Connectivity measures permit the network administrator to determine how much false data a compromised node is introducing into the network and what kind of impact it might have with respect to key generation and refreshment. Although compromise detection technology remains somewhat new, the characteristics of how it handles compromises should be examined closely.

In a balanced network, it can be challenging to determine whether a compromise has even occurred, much less the nature or extent of the compromising event. This is due to the fact that the management authority of the network is distributive and homogenous. In the proposed model, however, an N2 node is often used to generate a session key. Thus, the potential for detecting key compromise increases exponentially as the administrator can begin to narrow down the field of potential sites of compromise.

4.6.2 Node (N2) Compromise

Despite the fact that N2 nodes are better equipped structurally nodes to withstand penetration and infiltration attempts than N1, it is imperative to remember that no system
is entirely immune to the threats of network compromise. Therefore, it is important to consider how an N2 node might be captured and what happens when such an event transpires.

N2 nodes, even when small in number, can quickly compromise the immediate neighborhood and even the entire network if captured by a hostile force. The reason is that the N2 nodes contain valuable and sensitive information needed for session key generation. For this reason, system designers may consider the availability of resources to at least make the N1 and N2 nodes appear indistinguishable. Otherwise, hostile forces with sophisticated knowledge can target N2 specifically and compromise the network much more quickly.

4.7 Summary:

In this chapter, the proposed model for establishing communication between two neighboring nodes within a network was introduced and applied to a model by Younis et al. [42]. The new and old models’ formulations was discussed and the pros and cons of node compromising were analyzed. We discussed the various network connections that can be applied in order to establish a pair-wise key.
CHAPTER 5
CONCLUSION

Chapter 5 will present a theoretical approach towards a more suitable outcome in terms of establishing a pair-wise key protocol to aid in wireless sensor network security. A key feature of our approach as explored in this paper is that it exploits the availability of multiple transmission power levels at sensor nodes in terms of elevated security schemes.

Our proposed model focuses on establishing a key between two nodes while protecting data from being compromised. We have used a method that was previously proposed by [48], however, what makes our schema stand out is the use of a neighboring node as a gateway, which we will depend on in establishing a pair-wise key. Any pair of communicating nodes uses the base station as an intermediary for trusted communication and the key distribution is achieved through one-to-one communications, which does not scale well. Our approach splits the responsibility of key generation and distribution among multiple gateways and instead shares the key between two neighboring nodes, which provides scalability and prevents the manipulation of the network in the event of a compromise. Imposing a hierarchical structure extends a distributed, energy-efficient approach for ad hoc sensor network, based on their residual energy, and nodes will form clusters such that communication cost is minimized. A key feature of this approach is
We find that as the number of sensor network applications is expected to grow, especially in fields ranging from healthcare to warfare, it becomes increasingly important to ensure the security of data obtained from these networks. An essential component of any key-based security solution is managing the encryption keys in the system. In addition to guarding the network against attacks and localizing the effect of node compromises, key management in such vast resource-constrained networks should always be efficient and scalable.
BIBLIOGRAPHY


[8] Michael Steiner, Gene Tsudik, Michael Waidner. Key Agreement in Dynamic


[34] M. Steiner, G. Tsudik, and M. Waidner, Key agreement in dynamic peer groups, IEEE Transactions on Parallel and Distributed Systems, August 2000.


[42] Mohamed F. Younis, Senior Member, IEEE, Kajaldeep Ghumman, and Mohamed Eltoweissy, Senior Member, IEEE Location-Aware Combinatorial Key Management Scheme for Clustered Sensor Networks. IEEE Vol 17, No.8, August 2006


[49] Patrick Traynor, Heesook Choi, Guohong Cao, Sencun Zhu and Tom La Porta Networking and Security Research Center Department of Computer Science and Engineering The Pennsylvania State University, Establishing Pair-wise Keys in Heterogeneous Sensor Networks.

