Thesis for the Degree of Doctor of Philosophy

Locally Distributed, Energy Efficient, Scalable and Adaptable Key Management Solution for Clustered Sensor Networks

By

Syed Muhammad Khaliq-ur-Rahman Raazi

Supervised By

Prof. Sungyoung Lee, Ph.D.

Department of Computer Engineering

Graduate School

Kyung Hee University

Seoul, Korea

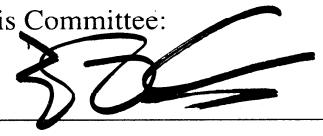
August, 2010

Locally Distributed, Energy Efficient, Scalable and Adaptable Key Management Solution for Clustered Sensor Networks

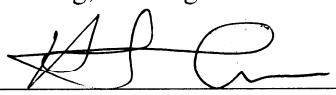
Syed Muhammad Khaliq-ur-Rahman Raazi

Submitted to
the Faculty of the Graduate School of Computer Engineering
in Partial Fulfillment of the Requirements
for the Degree of
Ph.D.

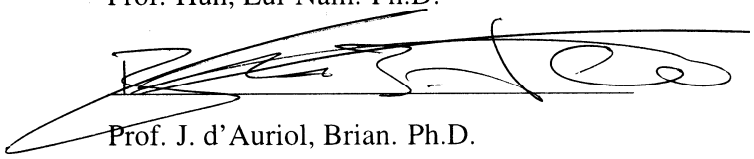
Thesis Committee:



Prof. Hong, Choong Seon. Ph.D.

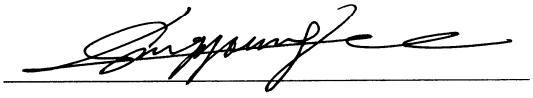


Prof. Huh, Eui-Nam. Ph.D.



Prof. J. d'Auriol, Brian. Ph.D.

Prof. Lee, Young-Koo. Ph.D.



Prof. Lee, Sungyoung. Ph.D.

Dedicated To

my parents Syed Muhammad Khalil-ur-Rahman and Syeda Shafiq Fatima, who work hard so that I can succeed & my loving wife Ayesha, who gives me peace of mind

Acknowledgments

In the name of Allah, the Beneficent, the Merciful

"Read! In the Name of your Lord, Who has created (all that exists), He has created man from a clot (a piece of thick coagulated blood) Read! And your Lord is the Most Generous, Who has taught (the writing) by the pen. He has taught man that which he knew not."

(Quran: Chapter 96, verses 1-5)

First of all, I would like to thank Allah, my lord, for providing me with everything that was necessary to successfully complete my Doctorate of Philosophy degree. Without his help, nothing is possible for me.

After that, I would like to thank my advisor Prof. Sungyoung Lee and my co-advisor Prof. Heejo Lee for guiding me in preparing this thesis, providing insightful comments and suggestions, criticizing my work to help me improve it and constantly encouraging me during my PhD studies.

Also, I would like to thank my MS advisors at Lahore University of Management Sciences (LUMS) Dr. Zartash Afzal Uzmi and Dr. Tariq Mahmood Jadoon for helping me to start active research. I would like to thank other faculty members of the Kyung Hee University especially Prof. Young-Koo Lee, Prof. Eui-nam Huh, Prof. Young-Jae Song, Prof. Man-Young Rhee, Prof. Salahuddin Muhammad Salim Zabir and Prof. Brian

J. d'Auriol for extending their support to me. I would like to thank all my colleagues especially Dr. Riaz Ahmed Shaikh and Dr. Syed Obaid Amin for their help and support.

Moreover, I would like to thank all my friends and colleagues at Kyung Hee University, especially Shoaib, Hidayath Mirza, Adil, Zafar, Asad, Zeeshan, Ozair, Faraz, Hassan Jameel, Dr. Tahir, Bilal, Dr. Uzair, Jehad Sarkar, Wang Jin, Weiwei, Dr. Donghai, Dr. Truc and Dr. Le Xuan, for their friendship and help in overcoming difficulties during my studies at Kyung Hee University. I would like give special thanks to the Korean members of ubiquitous computing lab. for their useful support in Korean language. Along with my friends, I would like to thank lab assistant of ubiquitous computing lab. Ms. Soungae Kim for taking care of administrative tasks and helping me at all times.

Finally, I would like to thank Korean government for providing me IITA scholarship for living expenses and Kyung Hee University for providing me President scholarship for tuition fees.

Syed Muhammad Khaliq-ur-Rahman Raazi

August 2010

Abstract

Wireless Sensor Networks (WSN) have proved to be useful in applications that involve monitoring of real-time data. There is a wide variety of monitoring and tracking applications that can employ Wireless Sensor Network. Characteristics of WSN, such as topology and scale, depend upon the application, for which it is employed. Security requirements in WSN vary according to application dependent network characteristics and characteristics of an application itself. Key management plays an important role in preventing adversaries from listening, disrupting or blocking private communications. Also, Key management is the most important aspect of security as other security modules depend on it. However, key management should not incur too much overhead in resource constrained WSN. This thesis aims to propose an energy-efficient key management framework, which prevents a WSN from having a single point of failure and can adapt according to application characteristics, for clustered wireless sensor networks. Its primary contribution lies in the development of unified, energy-efficient framework, called scalable and energy efficient key management framework.

WSN are susceptible to node capture and many network levels attacks. In order to provide protection against such threats, WSNs require lightweight and scalable key management scheme because the nodes are resource constrained and vary in number. Number of nodes in a WSN can be very high depending upon the application. Also,

effect of node compromise should be minimized and node capture should not hamper the normal working of a network. Moreover, WSN should not have single point of failure, which means that compromise of a single node should not compromise other nodes or a large number of nodes. Therefore, I present an EBS-based Key Management scheme called MUQAMI+ for large scale clustered sensor networks. I have distributed the responsibility of key management to multiple nodes within cluster, avoiding single point of failure and getting rid of costly inter-cluster communication. MUQAMI+ is scalable because the ratio of nodes, used for key management decreases drastically as the number of nodes in a cluster of WSN increases. Also, it is highly efficient in terms of key refreshment and revocation of compromised nodes.

Wireless body area networks (WBAN) consist of resource constrained sensing devices just like other wireless sensor networks (WSN). However, they differ from WSN in topology, scale and security requirements. WBAN have a small number of nodes tactically placed on a human body to support usability. When placed on a human body, many nodes fall in communication range of each other. Also, security requirements of WBAN differ from WSN because human intervention is possible or inevitable in many applications of WBAN. Moreover, WBAN are used to monitor biometrics, which can be used to generate key values. Due to these differences, key management schemes designed for WSN prove inefficient and unnecessarily complex when applied to WBAN. Also, monitoring of biometrics render key management of WBAN different from key management of WPAN. Therefore, I propose an energy-efficient and distributed key management scheme for WBAN called BARI+. BARI+ reduces key generation cost by using biometrics. Also, it has node revocation mechanism for applications, in which immediate human intervention can not be guaranteed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Background

WSNs differ from other distributed network systems in such a way that they have to work in real-time with given constraints, which include energy, storage, computation and communication. WSNs are mostly data centric and are used to monitor their surroundings, gather information and filter it [1]. A sensor network typically consists of a large number of sensor nodes working together to collect data and gather it in a central node, using wireless communications [2]. For detailed background on key management and wireless sensor networks, refer to [3].

Wireless Sensor Networks (WSN) are employed in various application areas [4],[5],[6]which include habitat monitoring [7], military surveillance and management, border monitoring and health care. Also, WSN are employed in industrial process control and smart indoor environments.

Habitat and environment monitoring are the most important applications of wireless sensor networks. In fact, these are the applications, for which wireless sensor networks were designed primarily. Numerous researches have identified these application areas for wireless sensor networks [8],[9],[10], challenging issues in them and their solutions [11],[12],[13].

Apart from monitoring environments and habitats, wireless sensor networks have been used for military and non-military surveillance. In monitoring applications, data is transferred to the base station at regular intervals. However, in surveillance applications, communication is mostly event-driven rather than being regular. Many researchers have identified research challenges in surveillance applications of wireless sensor networks [14],[15] and proposed their solutions [16],[17],[18].

Industrial process control is another application of wireless sensor networks [11],[19]. Industrial process control can be classified as indoor as well as outdoor depending upon the size of industrial plants. Sensor nodes can be deployed in those areas of an industrial plant, which can not be accessed easily and frequently.

Wireless Body Area Network (WBAN) is a very special scenario of wireless sensor networks and has some unique characteristics, due to which it should be treated separately. All sensor nodes, in a WBAN, are placed on a body, which is a human body in most cases [20]. Sensor nodes are very close to each other. Different sensor nodes have been designed for such sensor networks [21],[22] and separate protocols have been defined [23]. Also, researchers have been studying the effect of the presence of human body on such sensor networks [24],[25]. Apart from these differences, there is difference in application characteristics, which helps in key management for such networks[26]. Therefore, I have dedicated a separate section (section 1.7) in this chapter for discussing WBAN.

Apart from that, WSN can be used to assist human beings, provide them with a better lifestyle [27],[28],[29],[30] and help them in their problems. For example, sensor networks can be used for monitoring activities of elderly and ill people within their homes [31]. Although wearable sensor nodes are used to form WBAN, they can assist in making office and homes smart [32].

1.2 Wireless Body Area Networks

A wireless body area network (WBAN) is formed when sensor nodes are tactfully placed on human body to collect its biometrics or activities. Applications of WBAN include healthcare, lifecare and athlete examination. Healthcare includes care for inpatients especially those who are seriously ill, unconscious or under intensive care. Lifecare includes patients, who live their lives normally but may require medical care at any time. For example, lifecare facilities are useful in monitoring health of elderly people and pregnant women in real-time. Lack of timely medical care may cost some people their lives e.g. heart patients or high risk pregnant women. Also, WBANs are very useful in examining and monitoring an athlete's body.

Sensor nodes have less memory, computation and communication capabilities. Also, they have limited energy resources. Based on above properties, WBAN are classified into same category as WSN and treated in the same way when designing key management schemes. However, WBAN are different from usual WSN in many ways as discussed in [33].

Firstly, WBAN and WSN differ in scale. For WSN, number of nodes may be in thousands while WBAN consists of very few nodes, which may not exceed twenty. Obvious reason for this difference is usability. In humanware applications, sensor devices can be placed in watches, lockets or other wearable things. People may not agree to wear a lot of devices. If they do, it hampers their daily routine.

Secondly, nodes in WBAN are very close to each other as opposed to WSN. Nodes in WSN are scattered in large area like battlefield while nodes in WBAN are placed in small area i.e. a human body. Placing sensor nodes on a human body brings many of them in communication range of each other. Communication protocols have been designed keeping in mind such topology [23].

Table 1.1: Differences between WBAN and WSN

	WBAN	**WSN**
Scale	Small scale (Number of nodes may not exceed 20)	Large scale (Number of nodes may exceed even 1000)
Size of Operational Area	Very small (Size of human body). All nodes may be in communication range of each other	Spans large area like battlefields or natural habitat
Human Intervention	Possible rather inevitable in some cases	Not possible in most cases
Key Management Support from application	Yes, Sensor nodes need not generate random numbers	No

Thirdly, a compromised node can be physically removed in WBAN, which may not be the case in WSN because human intervention is not always possible in WSN. In WBAN applications, which are crucial for human life, it is essential to physically replace compromised nodes. For example, if there is only one node measuring a serious patient's heart rate, it must be replaced immediately. Since it is possible to physically remove a compromised node in WBAN, it is not efficient to use node eviction strategies in key management scheme.

Lastly, WBAN are used to measure biometrics from human body. Biometric values exhibit sufficient randomness properties and can be used to generate random numbers for cryptographic keys [34]. [34] uses "The last digit fluctuation method" to generate

random sequence from biometric data and extracts the least significant bit from every reading. Also, [34] proves that the least significant bit from every reading have sufficient randomness. According to [34], about n readings are required to generate an n bit key, which is viable because sensor nodes sense biometrics a lot more often than they relay its values to the central server.

There are two reasons for preferring physiological value based keying over pseudo-random number generators: Firstly, pseudo-random number generators require heavy computations as compared to physiological value based keying. Secondly, all random numbers are independent of each other in physiological value based keying as opposed to pseudo-random number generators. In pseudo-random number generators, a mathematical algorithm and a seed value are used to generate random numbers. If the algorithm and the seed value are exposed, the sequence of random numbers becomes deterministic. Also, obtaining truly random seed value is also a challenge. Phenomena that are measured in WSN applications may not have such randomness properties. WBAN can not be treated as a Wireless Personal Area Network (WPAN) because of the same reason. Some researchers use biometrics for key generation [35],[36]. Some of them argue that sensor nodes do not even need to exchange keys [37],[38],[39]. They rely on the assumption that two nodes can sense a biometric at the same time. After that, they apply error-correcting codes at both the communicating nodes. Apart from extra computations and time synchronization issues, this assumption imposes another constraint on the network i.e. Some nodes should be able to sense more than one biometric, which may not be practically possible. Also, such schemes do not take into account those nodes, which are not used for sensing biometrics.

Differences between WBAN and WSN are summarized in table 1.1. The only difference in security requirement of WBAN and WSN evident from table 1.1 is that a

Table 1.2: Differences between the security requirements of WBAN and WSN

	WBAN	**WSN**
Message Integrity	Required	Required
Node Authentication	Required	Required
Prevention from Eavesdropping	Required	Required
Node eviction through software	Not necessary	Required
Strategies to prevent routing attacks	Not required	Required
Prevention of attack propagation	Not required	Required

compromised node in WBAN scenario need not be evicted through software because human intervention is always possible. However, there is also difference between types of attack that can take place through compromised nodes in WBAN and WSN scenarios. In WBAN, one doesn't need to take care of routing attacks such as selective forwarding, wormhole and sinkhole attacks because many nodes have the cluster head node in their communication range. Nodes, which have very limited communication range, can communicate through one intermediate node. Moreover, due to the fact that WBAN are small scale networks, in which many nodes are in communication range of each other, one need not employ strategies to prevent attack propagation in WBAN. Table 1.2 outlines the differences between security requirements of WBAN and WSN.

From table 1.2, it is clear that the security requirements of WBAN are less complex than that of WSN. Also, from table 1.1 it can be learnt that more efficiency can be achieved in key management solutions if one exploits the characteristics of WBAN

applications while designing key management scheme for WBANs.

1.3 Motivation

In many applications of WSN, like military surveillance, security is the most important part. In other applications also, it is very important to conceal secret information from both active and passive adversaries. Moreover, an adversary can try to act like an authorized node to in order to extract important information from a legitimate node. Even if an adversary can not get to know the confidential information, it can try to disrupt communication or tamper with the messages, so that the wireless sensor network can not perform the task, for which it was deployed. In addition to that, an adversary may attack externally e.g. capture the node or jam the traffic signals. Key management is the most important part of WSN security. Apart from maintaining confidentiality, it assists other modules such as authentication, privacy and sometimes integrity.

In addition to being secure, WSN should also be cost effective because their batteries may not be recharged. This also limits their memory and computational power. So the WSN require security mechanisms that are also resource efficient [40],[41]. Highly effective security mechanisms such as Diffie-Hellman key exchange algorithm [42], RSA [43], TLS [44] and Kerberos [45] exist but they can not be applied to WSN paradigm because they are not resource efficient. However, key management in WSN is no less important than resource conservation. Many functions of wireless sensor networks depend on key management [46].

Firstly, key management framework for WSN should be scalable. Number of nodes in WSN vary depending upon the application for which the network is deployed. In applications like military surveillance, number of nodes can be very high. On the other hand, indoor applications of WSN can have a very few number of nodes. Key manage-

ment framework for WSN should be scalable so that its performance does not deteriorate with drastic increase or decrease in number of nodes in the target network.

Secondly, key management scheme for WSN should not be designed such that it relies heavily on a single node (in a network or in a cluster). In such cases the node, on which everything relies, becomes a single point of failure in the network and requires additional security. If a single point of failure is discovered, it becomes prime target for attackers. Moreover, if the single point of failure is compromised, whole network or whole cluster is compromised. Therefore, key management responsibility should be distributed among multiple sensor nodes. However, key management responsibility should not be distributed among all sensor nodes because WSN may have nodes, which can not afford to take part in key management.

Thirdly, key management framework for WSN should be able to not only work in different network topologies but also exploit characteristics of different applications. For example, number of sensor nodes decrease drastically when sensors are deployed on human body to form WBAN, which is a special type of WSN. Also, many sensor nodes in WBAN are within communication range of each other. Moreover, biometrics from the underlying application can be used for key management in WBAN.

Lastly, key management framework for WSN should not take assumptions that put additional constraints on WSN or sensor nodes. For example, LEAP+ [47] assumes that a WSN is safe during some initial time period, [39] assumes perfect time synchronization among sensor nodes when placed on human body and [48] assume that sensor nodes on human body can sense multiple biometrics. Such assumptions should be avoided when designing a key management framework for WSN.

1.4 Problem Statement

Key management is an important part of security in WSN. When talking about sensor networks, one also needs to consider resource constraints. However, key management schemes should have re-keying and compromised node revocation mechanisms. In current literature for WSN, there are state-of-the-art key management schemes, which are efficient and scalable [47]. Also, there are key management schemes that are distributed in nature so that they expose no single point of failure in a WSN [47], [49], [50]. Moreover, there are key management schemes that do not assume initial safe time period [49], [50]. All these features are important in key management schemes for WSN and no current key management scheme combines these features. Therefore, an efficient key management scheme for WSN is required that combines these features.

Literature for WBAN is not very different from WSN because many key management requirements of WBAN are same as that of WSN. For example, key management scheme for WBAN should have re-keying mechanisms, key management for WBAN should be distributed and energy efficient. Therefore, many state-of-the-art key management schemes of WSN can be applied in WBAN domain [47], [51]. However WBAN do have support for key management from the underlying application. Also, there are schemes that exploit such application characteristics [48], [39]. However, current key management schemes of WBAN either expose single point of failure [48] or they take assumptions, which put additional constraints on sensor network or sensor nodes. For example, some schemes for WBAN assume that sensor nodes have perfect time synchronization [39] and some schemes assume that sensor nodes can sense multiple biometrics [48], [39]. Key management schemes of WSN are overly complex and inefficient in WBAN because they do not exploit application and network characteristics of WBAN. Therefore, an efficient and distributed key management scheme for WBAN is

required that exploits application and network characteristics of WBAN and do not take assumptions, which put additional constraints on sensor network or sensor nodes.

1.5 Practical Considerations

It is important to consider practicality issues while trying to solve problems related to key management for wireless sensor networks. Key management is not the main task of any WSN. Key management is a service that facilitates completion of a main task in WSN. Therefore, key management cost should be kept to a minimum. Also, key management scheme should be compatible with existing frameworks designed for WSN [52]. Apart from the compatibility issue, there are other issues, which are discussed as follows: -

1.5.1 Wireless Sensor Networks

Limited battery power is the main issue of WSN. It is considered important when designing any protocol for WSN. Ideally, all battery power should be used for the main task of a WSN. However, adversary can easily drain energy from sensor nodes if there is a lack of sufficient security mechanisms. Also, key management scheme for WSN should not require sensor nodes to perform heavy computations because processors installed on sensor nodes have limited computation power.

Apart from that, key management scheme for WSN should also consider communication ranges of sensor nodes. In WSN, many nodes have very limited communication range. Therefore, key management scheme for WSN should not require all sensor nodes to communicate long distances. Also, sensor nodes also have very limited memory and key management scheme should not consume too much of it.

Moreover, key management scheme for WSN should not assume network to be safe

for any time period. WSN are deployed in critical scenarios and such assumptions can lead to disaster. Also, key management schemes for WSN should take into consideration node compromises because sensor nodes in WSN work unattended.

Key management overhead in WSN should be kept under practical limits. It is not necessary to provide absolute security in WSN. Adversaries try to eavesdrop on private communication or disrupt normal network operation but to a certain limit. After that, it becomes useless for adversaries to listen or disrupt private communication. Therefore, security should be provided up to the point, after which it becomes useless for an adversary to eavesdrop or disrupt network operation.

1.5.2 Wireless Body Area Networks

Just like in WSN, limited battery power is the main issue of WBAN. Although it is possible to recharge batteries of nodes used in WBAN, it is important to minimize energy drainage so that nodes do not require recharging very frequently. Like WSN, key management schemes of WBAN should not require sensor nodes to perform heavy computations or store large amount of data. Also, key management schemes for WBAN should be able to accommodate sensor nodes, which have very limited communication range.

Moreover, key management scheme for WBAN should not take assumptions, which put additional constraints on WBAN or sensor nodes. For example, number of nodes in a WBAN should be assumed within practical limits. Subjects may not allow too many nodes on their bodies.

In WBAN, maintaining confidentiality of personal information is a requirement that can not be argued. Therefore key management is essential in WBAN. However, key management schemes for WBAN should accommodate energy, memory, communication

and computation constraints.

1.6 Contributions

In this thesis, scalability and energy efficiency in key management for WSN is achieved by distributing key management responsibility locally within cluster and using EBS matrices. The primary contribution of this thesis is the proposition of unified, energy-efficient framework, called scalable and energy efficient key management framework for clustered wireless sensor networks as shown in Figure 1.1. The key management framework achieves scalability and energy efficiency and avoids single points of failure in WSN by locally distributing key management responsibility among sensor nodes. Also, it does not assume an initial safe time period. Also, the framework uses biometrics to manage keys in WBAN and does not require those sensor nodes, which can sense multiple biometrics. Also, it does not assume perfect time synchronization between sensor nodes. This framework has different components for key management in WSN and WBAN. Components of the framework are discussed below.

Key management for WSN: Key management for clustered WSN is addressed and a new scalable and locally distributed key management scheme MUQAMI+ is proposed for WSNs. MUQAMI+ has following unique features: -

MUQAMI+ avoids single points of failure in clustered WSN by distributing key management among multiple sensor nodes. By avoiding single point of failure, I reduce the risk of whole cluster or whole network being compromised due to the compromise of a single node.

MUQAMI+ distributes key management responsibility locally i.e. within a cluster. This reduces inter-cluster communication, which consumes a lot of energy, which is precious in sensor nodes.

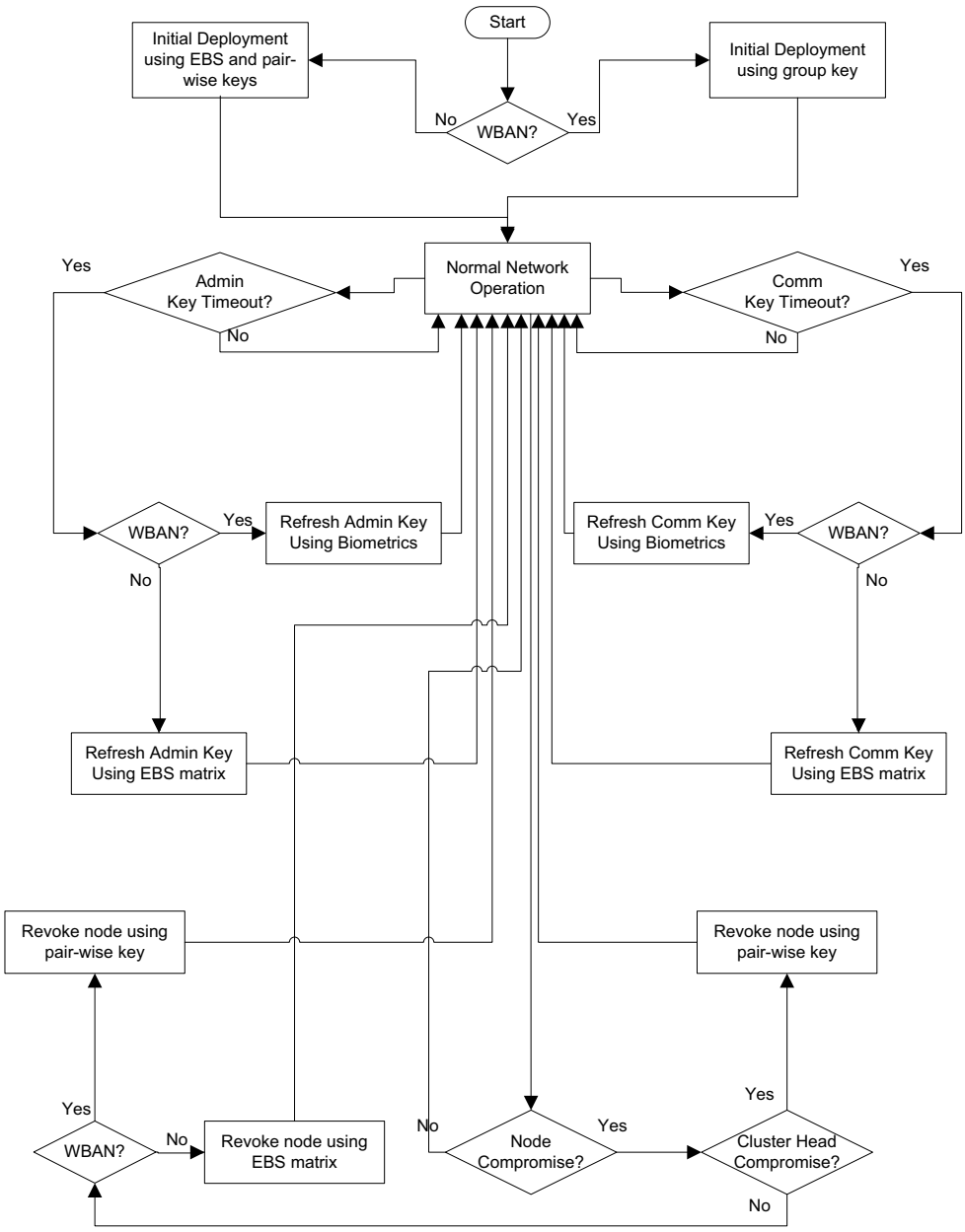


Figure 1.1: Locally distributed, energy efficient and scalable key management framework

MUQAMI+ uses EBS matrices to achieve scalability and efficiently revoke compromised sensor nodes. WSN remain energy efficient irrespective of number of nodes in the network.

In MUQAMI+, key management responsibilities can be transferred from one node to another with little cost, which allows sensor nodes to take turns in managing keys. Energy of nodes, with key management responsibility, will not be exhausted earlier than other nodes.

Due to the above mentioned features, MUQAMI+ overcomes the weaknesses and drawbacks of current work in key management for WSN.

Key management for WBAN: Key management for WBAN is addressed and a biometric based distributed key management scheme BARI+ is proposed for WBAN. BARI+ has following unique features: -

BARI+ uses biometric values from WBAN applications to generate random key values. Using biometrics for key generation reduces heavy computation costs associated with the generation of random values.

BARI+ distributes key management responsibility among multiple nodes in the network (all, which are in communication range of each other). This helps in avoiding single points of failure. Also, energy used for key management is spread evenly among nodes in the network.

BARI+ does not pose additional constraints on sensor nodes or the target network. BARI+ does not require sensor nodes to sense more than one biometric from human body. Sensor nodes, with capability of sensing multiple biometrics, cost more than the ones, which can sense single biometric. Also, BARI+ does not assume that the sensor nodes have perfect time synchronization.

The above mentioned features make BARI+ more viable for WBAN than existing

schemes that employ biometrics for key management. Moreover, BARI+ is designed keeping in mind scale, topology, network characteristics and application requirements of WBANs. This makes BARI+ more efficient than schemes, which are designed for generic applications of WSN.

1.7 Thesis Outline

Rest of this thesis is organized as follows: Chapter 2 presents related work. In the beginning of related work, security threats are discussed in detail. Then existing key management schemes for WSN are discussed in detail, starting from the simplest key management scheme present in literature and ending with the state-of-the-art solutions existing in current literature. In the end, I discuss key management schemes specific to WBAN domain.

Chapter 3 presents a scalable and locally distributed key management scheme for clustered WSN. Before presenting the scheme, chapter 3 summarizes drawbacks in existing work. Analysis and simulation results are presented in the end of chapter 3.

Chapter 4 starts with the summary of problems and issues in the existing work for WBAN. Then it proposes a distributed key management scheme, which exploits characteristics of WBAN for key management. In the end of 4, analysis and simulation results are provided.

In the end, Chapter 5 provides detailed security analysis of both schemes and then chapter 6 draws conclusions from this thesis and suggests future directions of this research.

Chapter 2

Related Work

2.1 Introduction

Whenever one thinks of the points that should be kept in mind while designing a key management scheme for wireless sensor networks, resource constraints (processing and memory capabilities) and energy constraint of the sensor nodes always come first. Otherwise, traditional key management schemes are very useful. It was due to the constraints of sensor nodes that always lightweight key management schemes are proposed for wireless sensor networks. However, maintaining required level of security in wireless sensor networks is also very important. Now I will discuss various key management schemes for wireless sensor networks proposed in the literature so far. [53] presented a very useful survey of key management schemes for wireless sensor networks. I will start from most simple key management solutions for wireless sensor networks and then discuss more complex ones later on. However, it is important to discuss assumed wireless sensor network system model and analyze vulnerabilities in it before discussing key management solutions that provide defense against those vulnerabilities.

2.2 Network Models and Assumptions

It is important to discuss assumptions and model of the system, for which I have proposed the key management schemes. Key management schemes, presented in this work, have efficient defence mechanisms against the vulnerabilities mentioned in Section 2.3. However, attack detection is out of the scope of this work. Readers interested in attack detection can refer to [54],[55],[56],[57] for further knowledge. In this section, I discuss various types of nodes present in wireless sensor network and wireless body area network and the way they communicate with each other.

2.2.1 Wireless Sensor Networks

Wireless sensor networks (WSN) consist of a command node connected to a number of sensor nodes, which can be grouped into clusters. I assume clustered sensor networks, in which cluster head node aggregates information from other sensor nodes and sends it back to the command node. Clustering can be based on some criteria like in [58],[59], where nodes do not know their locations. Sensor nodes relay their messages directly or indirectly, depending upon their communication ranges [60],[61]. There are many applications of WSN like soil moisture monitoring and battlefield monitoring, in which nodes do not change their location after initial deployment. It is important to minimize key management overhead in such applications also. I assume that all nodes, including the cluster heads, are stationary. Authors of other state-of-the-art schemes, like LEAP+ and SHELL, have also assumed the sensor nodes to be stationary. Communication range and physical locations of all nodes are known to the nodes at the higher levels. Figure 2.1 depicts the network architecture assumed in my scheme.

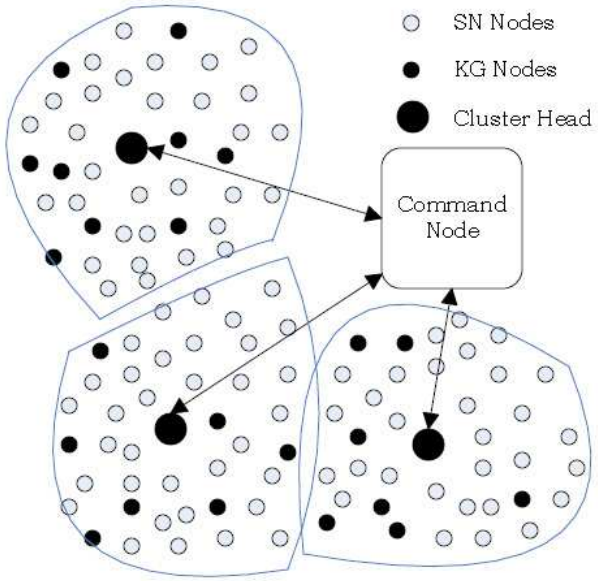


Figure 2.1: Architecture of Generic Clustered Wireless Sensor Networks

2.2.1.1 Command Node

Command node (CN) is at the top of the hierarchy of nodes in WSN. It gathers data or information from the underlying network through cluster head (CH) nodes. I assume that the CN is not constrained in energy, memory, communication or computation capabilities. Also, CN is connected with other networks, through which may be notified of node deployments, capabilities of the nodes deployed and values pre-loaded in them. CN has direct communication links with CH nodes. Other nodes send messages to the CN through their respective CH nodes. CN also uses the CH node to communicate with the individual sensor nodes.

Initial deployment and node addition phases of a WSN is initiated by the CN. Also, it helps new nodes in establishing their communication links. CN initiates key refreshment procedures if the decision is centralized. Also, CN plays an active role in compromised

node revocation especially if a cluster head node is compromised. It also has an active role when an ordinary sensor node is compromised because it spreads information about the compromised node in the network.

2.2.1.2 Cluster Head Node

Cluster head (CH) node is a sensor node, which gathers data from its cluster, aggregates the information or performs in-network processing, to reduce message size and number of messages, and sends useful data or information to the CN. The role of being cluster head requires more capabilities in a node as compared to the simple sensor nodes because a CH has to communicate with all nodes within its cluster and with the CN, which may be far from it. Also, the CH node has to store information related to nodes in its cluster and perform more computations as compared to other sensor nodes. Some sensor nodes, due to their restricted communication ranges or battery power, might not have direct communication link with the CH nodes. However, CH node can directly communicate with any node in its cluster. I assume that role of being a cluster head node can be transferred between nodes, which can bear the responsibility of being a cluster head node. We assume CN is available when responsibility of being CH is to be shifted from one node to another and there will be communication between CN and CH node when new CH node is selected.

CH nodes may play a part in the initial deployment of sensor nodes. Nodes, deployed in the node addition phases join through a CH node. If a sensor node joins through a CH node, CN sends identities of new nodes and/or initial key values, pre-loaded in them, to the concerned CH node. CH nodes play an important part in key refreshment and node revocation phases as they store information of all nodes and have direct communication links with all nodes. If decision for key refreshment and/or node revocation is not

centralized, CH nodes initiate these procedures in a wireless sensor network.

2.2.1.3 Sensor Node

Sensor (SN) nodes are at the lowest level in the hierarchy of nodes in WSN. The role of SN nodes is to sense their surrounding environment for phenomena, which they are required to monitor, and send useful information to the CN through their respective CH nodes. Depending on its communication range, a SN can communicate with its CH node directly or through some other SN node. Each node can be a member of only one cluster. If it falls within the boundaries of more than one cluster head nodes, it must choose one cluster for its membership.

SN nodes, deployed in initial deployment phase may be programmed to join the network through a CH node or communicate with the CN directly. However, nodes deployed in node addition phase join the network through a CH node. Role of SN nodes in key refreshment and node revocation phases depend upon key management scheme used in the network.

2.2.2 Wireless Body Area Networks

Scenario of WBAN is such that there are few sensor devices, which are capable of measuring biometrics related to human body. These devices are tactically placed on a human body in such a way that they do not hamper daily routine of the human being. Also, there is a personal server, which can be a laptop or a hand held device. The personal server and all the sensor nodes form a wireless body area network (WBAN). Sensor nodes measure biometrics and forward them to the PS. In turn, the PS relays this information to medical server, through the internet. Each WBAN is associated with only one body. Multiple WBANs are associated with one central MS. System architecture, as per my assump-

tions of WBAN, is shown in Figure **??**. I assume that all nodes are pre-loaded with node identities and relevant keys before deployment. For critical scenarios and rapid deployments, sensors that are targeted for the same WBAN and the associated PS can be grouped together in advance. I assume that the PS and all sensor devices are constrained in energy because they use rechargeable batteries.

2.2.2.1 Medical Server

Medical Server (MS) is a central node, which stores information gathered from multiple personal servers (PS). Also, it processes that information and generates alerts, for concerned personnel e.g. doctors or nurses, based on the output. Also, authorized people can access required information from the MS. Different data mining strategies are employed on medical server to make use of data from health care applications [62]. MS is not constrained in energy, computation, communication or memory. MS is connected with the PS through an external network. Security of the network, through which MS and PS are connected is out of scope of this work. This external network may be the internet. MS communicates with other sensor nodes in the network through the PS.

MS initiates the initial deployment phase and node addition phase of a WBAN by indicating new deployments to PS nodes, which then expect new nodes. Also, it has plays an active role in network recovery if a PS node is compromised.

2.2.2.2 Personal Server

Personal Server (PS) gathers information from sensor nodes, placed on human body, processes it and sends it to the MS. Personal applications that run on PS can also handle a person's biometrics information [63]. PS is a hand-held or a laptop class device, which can communicate with the MS through an external communication channel. In some

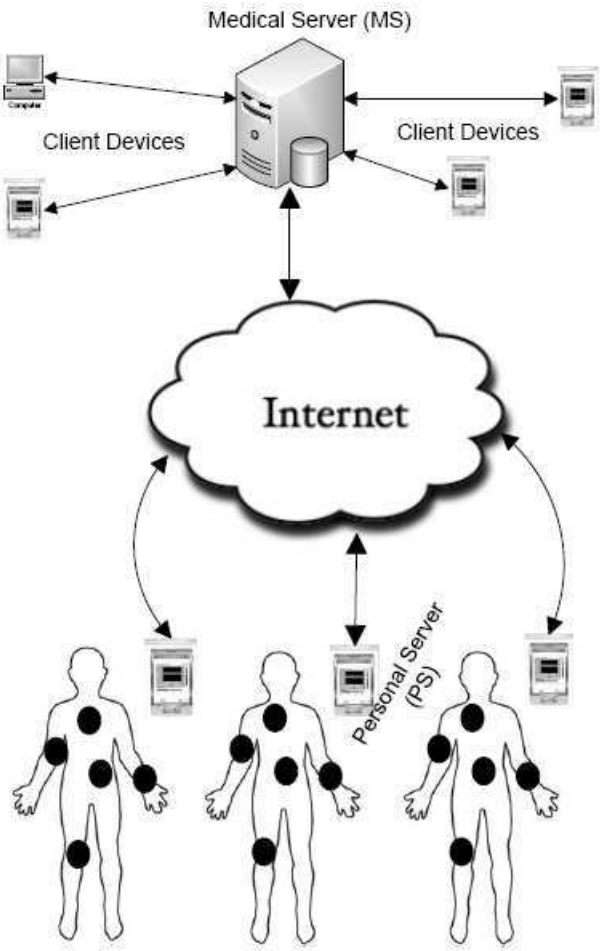


Figure 2.2: System Architecture of Wireless Body Area Networks

cases, PS may be a sensor node. PS nodes are more capable than ordinary sensor nodes. However, they are also battery powered devices and has energy constraint. Some sensor nodes, due to their restricted communication ranges or battery power, might not have direct communication link with the PS node. However, PS node can directly communicate with any node in its WBAN.

PS carries out initial deployment and node addition phases. PS is deployed before ordinary sensor nodes, which join the network through the PS. Also, it initiates key refreshment phase and revokes a compromised sensor node.

2.2.2.3 Sensor Node

Sensor (SN) nodes are at the lowest level in the hierarchy of nodes in WBAN. Role of SN nodes is to sense biometrics from human body and send useful information to the PS, with which it is associated. Depending on its communication range, a SN can communicate with the PS node directly or through some other SN node.

SN nodes may assist in initial deployment or node addition phase if a newly deployed SN node can not communicate with the PS node directly. Role of SN nodes in key refreshment phase depends upon key management scheme used in the network.

2.3 Vulnerability Analysis and Attack Vectors

The main goal of a key management scheme is to ensure confidentiality of information. Also, keys can be helpful in authenticating legitimate nodes. An adversary may try to crack secret key and extract confidential information from the messages exchanged between communicating nodes. If keys are used for authentication purposes, adversary may try to act as a legitimate node and try to extract confidential information from other

nodes. While trying to crack a secret key, adversaries try to learn message patterns [64] and guess the secret key. Also, they try to save some encrypted messages, which they can replay later on. In order to prevent adversaries from guessing secret keys, it is important to refresh keys at appropriate time intervals. Time intervals depend upon frequency of communication and frequency of key usage.

Apart from trying to crack secret information, adversary can also harm a sensor network in several other ways. It can try to jam wireless signals of a sensor networks. Also, it can try to create noises and disrupt communication. In other words, adversary can carry out denial-of-service attacks. Apart from that, an adversary can try to drain sensor nodes' energy by initiating bogus messages or replaying old messages. Although many of such attacks, like signal jamming, can not be handled by key management schemes, but I think it is important to list them at least. Readers can refer to the security mechanisms at physical layer to find remedy of jamming attacks. [65] classifies all security attacks in wireless sensor networks in four classes: interruption, interception, modification and fabrication. Interruption is when a communication link in interrupted. Interception takes place when a sensor node or its data is compromised. In modification, adversary gains access and tampers with the data. Finally, fabrication takes place when an adversary injects false data into the network. [66], [47], [67] and [68] are also very helpful resources for studying about security attacks that can take place in wireless sensor networks.

Broadly speaking, attacks on wireless sensor networks can be classified in two categories: outsider attacks and insider attacks. In outsider attack, adversary is not a part of the network. For example, jamming attack is carried out on physical layer of the sensor nodes by a node, which is not part of the sensor network so it is classified as an outside attack. In insider attack, an insider node (a node from within the network) is compromised through node tampering or through a weakness in its system software. [65] is an

important resource to study categorization of attacks in WSN.

In this section, I discuss vulnerabilities of WSN and WBAN and their related attack vectors. WSN and WBAN share all the vulnerabilities except the ones, which involve routing. Routing attacks do not affect WBAN as much they affect generic WSN applications because in WBAN most nodes are just one or two hops away from the sink node. However, I list the vulnerabilities and attack vectors associated with WSN and WBAN separately. In this regard, I present vulnerabilities of WSN and their related attack vectors: -

2.3.1 Wireless Sensor Networks

Although it is more difficult for an adversary to compromise a node higher in hierarchy, it is important to discuss all possibilities of attack. In WSN, main vulnerabilities include passive listening, illegitimate packet injection, illegitimate node injection, communication disruption and node capture [52]. All attacks take place because of these basic vulnerabilities: -

2.3.1.1 Passive Listening

Passive information gathering is the most common vulnerability in wireless networks, which do not involve encryption/decryption and key management. In this case, presence of an adversary is not evident to the target WSN. If information is not encrypted, an adversary can listen to the communication passively. Even if information is encrypted, an adversary can analyze the communication patterns of a sensor network and cause harm to the network. For example, if all packets are routed through a single node, it can cause compromise of that single node. This is one reason why cluster head nodes have more security as compared to other nodes in clustered sensor networks. Apart from that,

traffic analysis attacks also highlight the need for refreshing keys at regular intervals. Following are the attack vectors associated with the vulnerability of passive listening: -

1. Adversary can listen to the private communication between two sensor nodes so that confidentiality of information is breached.

2. Even if messages are encrypted, adversary can analyze traffic patterns, which may lead to compromise of an important node.

3. Adversary can carry out cryptanalytic attacks to reveal secret keys.

2.3.1.2 Illegitimate Packet Injection

Without proper authentication and integrity checking mechanisms, it is possible for an adversary to affect communication between sensor nodes by injecting illegitimate packets in the network. Illegitimate packets may include false packets or packets, which have already been communicated to the receiving node. Adversary can modify contents of a message before it gets to the receiver. In order to do so, adversary need not capture a node or introduce an illegitimate node. Adversary can use a compromised key and use a powerful radio transceivers from a distance to exploit this vulnerability.

Adversary can spoof link layer acknowledgement after overhearing packets (**Acknowledgement Spoofing**). Suppose there are two nodes A and B. Node A wants to send some data to the base station through node B but node B is dead. A compromised or outsider node E overhears the initial message sent by node A and spoofs an acknowledgement to node A at link layer. Based on spoofed acknowledgement, node A starts forwarding its packets to the base station through node E. After that, node E drops some or all packets forwarded to it by node A.

An attacker can use compromised nodes or outside malicious nodes to play with

routing information in such a way that it creates routing loops, attracts or repels network traffic, alters source routes or generate false error messages (**Spoofed, Altered or Replayed Routing Information**). Apart from other hazards, this type of attack cause large network delays and also drain the battery power of sensor nodes very quickly. Also, an adversary can send a HELLO packet or replay a routing protocol's HELLO packet with more signal strength (**Hello Flood Attacks**). As a result, each of the other sensor nodes thinks that the malicious node is its neighbour. Then the malicious node can advertise a low latency link creating a wormhole. Also, sensor nodes waste their energies in responding to HELLO floods.

In short, following are the attack vectors are associated with this vulnerability: -

1. Adversary can inject false application/routing information in the network to cause application malfunction e.g. **Acknowledgement Spoofing**.
2. Adversary can inject altered data/routing packets in the network.
3. Adversary can access confidential information and pass it to an enemy.
4. Adversary can inject large number of packets in the network to cause node outage or denial-of-service.
5. Adversary can modify application/routing information to affect WSN operation.
6. Adversary can replay packets to cause node outage or routing issues.
7. Adversary can send routing protocol's HELLO packets with more signal strength to cause a **hello flood attack**.
8. Adversary can analyze traffic to determine nodes' responsibilities.

2.3.1.3 Illegitimate Node Introduction

In WSN, it is really important to have proper authentication mechanisms, based on secret keys, so that illegitimate nodes can not become part of a network. Malicious node (**False Node**) can be introduced in the network by an adversary. This malicious node tries to inject malicious data and attract other nodes to send data to it. For example, it can advertise shortest route to the base station so that other nodes route their packets through it in order to save energy. Also, an illegitimate node can attract network traffic (**Sinkhole Attack**). After that, the malicious node can carry out **Selective Forwarding** on the traffic. Sinkhole and selective forwarding attacks are most effective if the illegitimate node is near the base station.

By introducing illegitimate node, an adversary can cause **Sybil Attacks**, in which a malicious node presents multiple identities in the sensor network. In doing so, it can either steal other nodes' identities or it can try to fabricate new identities itself. Basically, sybil attacks reduce effectiveness of fault tolerant schemes like distributed storage. Also, sybil attacks can affect routing algorithms. For example, it can cause a routing algorithm to determine two disjoint paths, which are not disjoint in reality. Also, adversary can launch **Wormhole Attacks**, using two distant malicious nodes, which can communicate with each other, through an out-of-band communication channel, which is invisible to the underlying sensor network. One of the malicious nodes is placed near the base station and the other one is placed near the sensor nodes, which generate data. Using this low latency link the malicious node, which is placed near the data generating sensor nodes, convinces data generating nodes that it is just one or two hops away from the base station. This can cause sinkhole in the network. Also, this can create routing confusion especially in malicious node's neighbours, who might think that the other malicious node, near to the base station, is their neighbour.

In summary, we can enumerate attack vectors associated with this vulnerability as follows: -

1. Illegitimate node or **false node** can inject false information in the network.
2. Illegitimate node can inject a large number of packets in the network. If such packets are entertained by other nodes, it drains their energy. Otherwise it causes denial-of-service attack.
3. Illegitimate node can breach information confidentiality by passing private information to a foe.
4. Illegitimate node can modify information being transferred from source node to sink node especially during in-network processing. Also, it can cause problems in routing.
5. Illegitimate node can suppress information being transferred from one node to another to cause application malfunction or routing issues e.g. **Selective Forwarding**.
6. Illegitimate nodes can replay packets that have already been transmitted from one node to another. This can drain other sensor nodes' energy or create routing issues in the network.
7. Illegitimate node can attract other nodes to route their packets through it (to cause a **sinkhole**) so that it can modify/suppress information.
8. Illegitimate node can spoof acknowledgements of dead or absent nodes.
9. Illegitimate node can present multiple identities in a WSN i.e. carry out a **sybil attack**.

10. Two illegitimate nodes can be introduced in a WSN to cause a **wormhole** in the network.

11. An illegitimate node can cause a hello flood attack by sending routing protocol's HELLO packet with more signal strength.

2.3.1.4 Node Capture/Compromise

In most WSN applications, sensor nodes have to work unattended. This causes another vulnerability in the network. In this vulnerability, an adversary can capture a legitimate sensor node physically. Also, a legitimate sensor node can be compromised through holes in system software. After compromise, adversary gains access to data, information and cryptographic keys stored in the node. Adversary can also cause the node to malfunction and generate inaccurate data. Also, an adversary can remove a captured node from the sensor network or exhaust the node's energy (**Node Outage**). Following is the list of attack vectors are associated with this vulnerability: -

1. Adversary can access private information stored on the compromised sensor node and deliver it to a foe.

2. Adversary can inject false application/routing information in the network. This can cause application malfunction or routing problems.

3. Adversary can access secret keys stored on the compromised node. Adversary can pass secret key to an illegitimate node or use it to query other legitimate nodes in the network.

4. Adversary can turn off the compromised node so that it can no longer take part in the task assigned to the network.

5. Adversary can inject huge amount of traffic in the network to cause **node outage** or denial-of-service.

6. Adversary can modify information to cause application malfunction or routing problems in the network.

7. Adversary can suppress routing or application information being sent from one node to another.

8. Adversary can replay application/routing information to cause node outage or routing problems.

9. Compromised node can be used to attract nodes for routing their packets through it so that adversary can modify/suppress information.

10. Adversary can spoof acknowledgements of nodes, which are not present.

11. Adversary can use a compromised node to cause a sybil attack.

12. Compromised node can be used to carry out hello flood attack.

13. Compromised node can collude with another compromised node or an illegitimate node to cause wormhole in the target network.

2.3.1.5 Communication Disruption

Like other wireless networks, WSN are also vulnerable to disruption of communication on physical layer. An adversary can carry out DoS attack (**Denial of Service Attack**) by disrupting communication between sensor nodes. Typically, DoS attacks occur at the physical layer of wireless sensor networks. Radio Jamming is a classical example of a DoS attack. There are two attack vectors associated with this vulnerability

1. Adversary can jam radio signal, through which sensor nodes communicate e.g. **Denial of Service Attack**.

2. Adversary can introduce a lot of noise in the channel, through which sensor nodes communicate.

2.3.2 Wireless Body Area Networks

Wireless Body Area Networks (WBAN) have many similarities with WSN because WBAN is also a network of sensors. However, most of the nodes in WBAN are in communication range of each other. Those, with very limited communication ranges, are only one or two hops away from the PS node. Therefore, attacks that involve routing do not apply to WBAN scenario. Vulnerabilities of WBAN are same as that of WSN but there are differences in the related attack vectors. Vulnerabilites and attack vectors of WBAN are listed as follows: -

2.3.2.1 Passive Listening

Just like in WSN, passive information gathering is the most common vulnerability in WBAN, which do not involve encryption/decryption and key management. If information is not encrypted, an adversary can listen to the communication passively. Even if information is encrypted, an adversary can analyze the communication patterns of a sensor network and cause harm to the network. Following are the attack vectors associated with the vulnerability of passive listening: -

1. Adversary can listen to the private communication between two sensor nodes so that confidentiality of information is breached.

2. Adversary can carry out cryptanalytic attacks to reveal secret keys.

2.3.2.2 Illegitimate Packet Injection

Without proper authentication and integrity checking mechanisms, it is possible for an adversary to affect communication between sensor nodes by injecting illegitimate packets in the network. In short, following are the attack vectors are associated with this vulnerability: -

1. Adversary can inject false application information or altered data packets in the network to cause application malfunction.
2. Adversary can access confidential information and pass it to an enemy.
3. Adversary can inject large number of packets in the network to cause node outage or denial-of-service.
4. Adversary can modify application information to affect WBAN operation.
5. Adversary can replay packets to cause node outage.

2.3.2.3 Illegitimate Node Introduction

In WBAN, it is really important to have proper authentication mechanisms, based on secret keys, so that illegitimate nodes can not become part of a network. Attack vectors, associated with the vulnerability of Illegitimate Node Introduction can be enumerated as follows: -

1. Illegitimate node or **false node** can inject false or altered information in the network.
2. Illegitimate node can inject a large number of packets in the network. If such packets are entertained by other nodes, it drains their energy. Otherwise it causes denial-of-service attack.

3. Illegitimate node can breach information confidentiality by passing private information to a foe.

4. Illegitimate node can suppress information being transferred from one node to another.

5. Illegitimate nodes can replay packets that have already been transmitted from one node to another. This can drain other sensor nodes' energy.

2.3.2.4 Node Capture/Compromise

In In WBAN, human intervention is possible most of the time. In fact, it is inevitable in many cases. However, human intervention can not be guaranteed in all WBAN scenarios. Therefore, node capture is a possible vulnerability in WBAN scenarios. Following is the list of possible attack vectors are associated with this vulnerability of WBAN: -

1. Adversary can access private information stored on the compromised sensor node and deliver it to a foe.

2. Adversary can access secret keys stored on the compromised node. Adversary can pass secret key to an illegitimate node or use it to query other legitimate nodes in the network.

3. Adversary can inject false or modified application information in the network. This can cause application malfunction.

4. Adversary can turn off the compromised node so that it can no longer take part in the task assigned to the network.

5. Adversary can inject huge amount of traffic in the network to cause node outage or denial-of-service.

6. Adversary can suppress information being sent from one node to another.

7. Adversary can replay application information to cause node outage.

2.3.2.5 Communication Disruption

WBAN are also susceptible to communication disruption occurring at physical layer Like other wireless networks. Just as the case in WSN, there are two attack vectors associated with this vulnerability

1. Adversary can jam radio signal, through which sensor nodes communicate e.g. **Denial of Service Attack**.

2. Adversary can introduce a lot of noise in the channel, through which sensor nodes communicate.

2.4 State-of-the-art Research in WSN

Apart from the required level of security, key management schemes designed for wireless sensor networks should also cater for constraints related to sensor nodes. Apart from limited bandwidth, memory and computation capabilities, sensor nodes do not have any prior knowledge regarding their deployment. Limited transmission range and limited battery life also add to the constraints. Limited battery life is the primary reason why asymmetric key management strategies are not considered suitable for wireless sensor networks. Asymmetric key management schemes perform intense mathematical calculations, drains a lot of energy from sensor nodes.

Many sensor nodes can only transmit up to short distances. Therefore, some sensor network data collection techniques employ in-networking processing [69], [70] and

[52]. In in-network processing, all nodes send their data to a few nodes, which aggregate messages and transmit only processed information towards the command node. In order to avoid unnecessary communication, some schemes in wireless sensor networks require nodes to overhear messages from other nodes [66] and [71]. It is fruitful if key management schemes support in-network processing and message overhearing.

Single Group Key for a Network: It is by far the simplest key management scheme used for wireless sensor networks. In this case, a single key is loaded into every sensor node before deployment. All sensor nodes communicate using that single key. Sometimes, a single group key is used for a cluster rather than for whole network [72],[73]. This scheme is very lightweight in terms of memory, computation and communication requirements. It is also flexible and scalable but at the same time, it is also very vulnerable. If a single key is used for a long time, chances of cryptanalytic attack on the key gets higher and it is easier for an adversary to compromise the key. In this scenario, if a node is compromised or the key is revealed in some other way whole network is compromised. There is no way one can refresh the key or revoke the compromised sensor node from network and retain rest of the network.

Pair-wise Key Establishment: Establishing pair-wise key between every pair of nodes in a sensor network is the most secure key management scheme for wireless sensor networks. Every node is preloaded with a key for communication with every other node. For instance, if there are n nodes in a network every node will have n-1 keys stored in its memory. This scheme is possibly the most secure for wireless sensor networks. Pair-wise key establishment not only provides confidentiality and authenticity in a network but also provides efficient mechanism for revocation of a compromised sensor node in the network. However, this scheme is not at all efficient in terms of scalability and memory requirements. If the number n becomes too large as in many applications

of wireless sensor networks, this scheme becomes impractical. Also, communication between every pair of sensor nodes is not necessary in wireless sensor networks.

Random Pair-wise Key Establishment: [74] argue that all nodes in a sensor network need not share pair-wise keys. In their approach, two nodes share a pair-wise key with some probability p and p must be chosen carefully in order to keep the network connectivity up to a desired level. Also, node revocation does not need to involve the base station. A node's status in the network depends upon the consensus among nodes, with which it communicates. If a certain number of nodes, with which it communicates, say that node A is compromised, all of them will terminate their communication with node A. Although this scheme works well for small networks, it does not scale well enough if network size becomes too large.

Trusted Key Distribution Center (KDC): In this approach, drawbacks of pair-wise key management are mitigated by storing all pair-wise keys in a key distribution center. This key distribution center can be the base station or a cluster head node in clustered sensor networks. Although this approach is secure and resilient against node capture and node replication, this approach is also not scalable. This is because every pair of nodes has to obtain keys from the trusted base station for every session. Apart from the communication overhead introduced in this approach, links around the base station may become overloaded. If trusted KDC is a sensor node e.g. a cluster head node in clustered sensor networks, then its memory requirements increase manifold. Also, it must have far better energy and communication capabilities. In addition to all that, the trusted key distribution center becomes a single point of failure especially if it is one of the sensor nodes.

Random Key Pre-distribution Scheme: In wireless sensor networks, it is not necessary that keys are established among every pair of sensor nodes. For a wireless sen-

sor network to work, it is important that every sensor node gets sufficient bandwidth and neighbouring nodes, who can relay its messages to the base station through various paths. For example, if node A has 15 nodes in its neighbourhood, it can establish pair-wise keys with only 4 of them and those 4 neighbouring nodes can provide node A distinct routes to the base station, then node A does not need to establish pair-wise keys with rest of the 11 nodes. Random key pre-distribution scheme was proposed by [75]. In the first phase of their scheme, a key ring of K keys and their identifiers is stored in the memory of each node prior to deployment. Every pair of nodes shares a key with some probability. In discovery phase, every node broadcasts its key identifiers and challenges to find those nodes, with which it shares a key. If some keys are left unused after the discovery phase, they can be used to establish keys between nodes, who do not share a common key. For example, node A shares a key x with node B and node B shares another key y with node C while nodes A and C do not share a key. If node B has a key z, which it does not share with any node, it can send key z to both node A and node C so that they can communication with each other using z. In this scheme, there are group keys that are shared between the base station and all other nodes. In order to revoke a compromised sensor node, the base station compiles the list of keys known to the compromised node, uses a group key to sign the list and broadcasts it into the network using another one. Upon receiving the list, all nodes delete the keys, which are known to the compromised node, from their memory. Apart from the fact that shortest path to the base station might not be established in this scheme, another drawback is that node revocation might cause many other links, which use one of the deleted keys, to break.

Q-Composite Random Key Pre-distribution scheme: Q-Composite random key pre-distribution scheme, which was an improvement to the random key pre-distribution scheme, was proposed by [74]. In Q-Composite scheme, two sensor nodes must share

at least q number of keys if they want to establish a link between themselves. In this way, two linked nodes will have other keys for communication if one of the keys is compromised. In this case, size of the random key pool need to be reduced to maintain the probability that two nodes share q common keys. This poses another security problem: adversary will need to compromise only a few sensor nodes to compromise most of the keys. Also, keys can not be refreshed when they are only pre-distributed.

Multi-path Key Reinforcement Scheme: In basic random key pre-distribution scheme, multiple nodes may share more than one key. In this case, if one node is compromised, there is a chance that links between other non-compromised sensor nodes may also be compromised. In order to solve this problem, [74] proposed that keys, which are used for communication on links between other non-compromised nodes, should be refreshed but not through already established link. For this purpose, they use multiple disjointed paths between two nodes. If two nodes A and B share a common key k, and they have h disjointed paths between them, node A generates h random values and sends each one of them to B through a separate disjointed path. Then both nodes A and B compute a key k' using key k and all h random values. Even if a node in a path is compromised, adversary will not know k' and k can be refreshed through k'. In order to keep the chances of eavesdropping to a minimum, size of disjointed paths should be kept small. Apart from increased network communication, this scheme also increases the computation overhead of sensor nodes by requiring them to generate random values, which require a lot of energy.

Polynomial Pool-based Key Pre-distribution: In polynomial Pool-based key pre-distribution scheme [76], a setup server generates one t-degree polynomial for each sensor node. These polynomials hold the property $f(x, y) = f(y, x)$. For example, if node i receives a polynomial $f(i, y)$ and node j receives a polynomial $f(j, y)$, they can com-

pute a common key using identity of the other node. This scheme is scalable. However, whole network is compromised if t nodes are compromised. Also, its memory requirements increase with increase in the value of t.

Grid-based Key Pre-distribution: This approach is similar to the polynomial based key pre-distribution approach. In this approach, a matrix is stored in each node's memory. If two nodes i and j want to establish a pair-wise key for communication, they must have a common row or column in the matrix. If none of the rows or columns matches, then they must find alternate path to each other in path key establishment stage. This scheme offers greater probability of key establishment as compared to the random pair-wise key establishment scheme. This scheme reduces communication and computation overhead but increases the storage overhead.

Public Key Cryptography in Wireless Sensor Networks: In previous sections, I discussed that like other traditional key management schemes, public key cryptography can not be used in wireless sensor networks due to highly sophisticated computations involved in it. Contrary to this point of view, many researchers argue that the use of public key cryptography on wireless sensor networks can not be ruled out completely [77] especially the Elliptic Curve Cryptography (ECC), which has been used in wireless sensor networks recently [78], [79], [80]. Also, public key schemes have been used on 8-bit processors [79]. ECC can provide same level of security as that of RSA with much smaller key. According to [79], 160-bit ECC key has the same level of security as that of 1024-bit RSA. Also, the difference in the number of bits is not constant because 224-bit ECC has the same level of security as that of 2048-bit RSA key. ECC based public keys have been used in TinyOS [78], an operating system developed specifically for wireless sensor networks.

Some schemes provide hybrid approach for key management i.e. they mix both sym-

metric and asymmetric key management approaches for providing security in wireless sensor networks [81], [82]. LSec [83] also uses hybrid approach for key management in wireless sensor networks. In the first phase, they perform authentication and authorization using symmetric keys. In the second phase, keys are distributed using random secrets. This is performed using asymmetric cryptography.

SHELL: SHELL [49] is location-aware combinatorial key management scheme designed for clustered sensor networks. I will discuss SHELL and the rest of the schemes in a bit more detail because they are state-of-the-art solution of key management so far in the literature. SHELL assumes large scale sensor networks with clusters sizes of the order of hundreds of nodes. SHELL uses a small number of keys to manage large sensor networks using combinatory. SHELL employs EBS system of matrices [84] to use small number of keys for large networks. In addition to using small number of keys for large networks, SHELL also gets rid of single point of failure by using neighbouring cluster heads for key management. SHELL targets sensor networks that are hierarchical i.e. a cluster head node manages a large number of sensor nodes and a base station manages multiple cluster head nodes. In other words, this scheme is suitable for networks, which support in-network processing of information. Also, this scheme supports overhearing of messages as one key is known to a large number of nodes.

SHELL assumes that the cluster head nodes can broadcast messages to all the sensor nodes in its cluster. Also, the cluster head node can reach all nodes in its own cluster. However, if a cluster head node wants to communicate with some node, which is not in its cluster, it has to go through the neighbouring cluster head node. In short, cluster head nodes have more communication, computation, storage and power capabilities as compared to other sensor nodes. Base station or the command node has minimal involvement in this key management protocol. Another important assumption taken in SHELL is that

Table 2.1: Example of an EBS matrix

	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
K_1	**1**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
K_2	**1**	**1**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**
K_3	**1**	**0**	**0**	**1**	**1**	**0**	**1**	**1**	**0**	**1**
K_4	**0**	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**
K_5	**0**	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**1**	**1**

two compromised nodes can't come to know the location of each other i.e. they can not launch a coordinated attack. Also, two compromised nodes can't communicate through an out of band communication channel. Lastly, attacker does not know memory contents of a sensor node before deployment. EBS system of matrices is used by SHELL and another state-of-the-art key management scheme; I think it is important to discuss EBS system of matrices briefly.

Table 2.1 shows an example of EBS matrix. Size of an EBS matrix depends upon the number of nodes and the number of keys used to manage those nodes. In EBS matrix, number of columns is equal to the number of nodes in a cluster. Number of rows is equal to the number of keys used to manage those nodes. Total number of keys in a network is $k + m$. Out of these $k + m$ keys, every sensor node knows a distinct set of k keys i.e. set of keys known to one of the sensor nodes can not be exactly identical to the set known to some other sensor node.

If a node is compromised, set of m keys, which are not known to the compromised node, are used to refresh the k keys known to the compromised node. Suppose that in table 2.1, Node $N1$ is compromised. Set of k keys known to $N1$ is $K1$, $K2$ and $K4$. If

the managing node generates new values of $K1$, $K2$ and $K4$, encrypts each one of them in $K3$ and $K5$ separately, and broadcasts in the cluster, all of the nodes will be able to decrypt the message except the node $N1$.

It is very important to note that the number of nodes supported by $k + m$ keys grows exponentially with the values of k and m. Values of k and m can be adjusted according to the network and its security requirements. Higher value of m results in higher security but with increased overhead.

Initially, each sensor node is pre-loaded with a discovery key K_{sg} and two other keys KS_{CH} and KS_{Key}. K_{sg} is recomputed with one-way hashing function, such as SHA1 [85] or MD5 [86], stored in the node. The one-way hashing function is only known to the sensor node and the command node. K_{sg} is used to recover the network if the cluster head node is compromised. KS_{CH} and KS_{Key} are used for initial key distribution. In a cluster head node, key K_{gc}, which is used for communication between the cluster head node and the command node, is pre-loaded along with key K_{sg} of all nodes that lie in its cluster. Gateways can also communicate between themselves using another type of key provided by the command node. Command node generates the key, used for communication between the cluster head node, and renews them at regular intervals.

In SHELL, cluster head node is responsible for the formation of EBS matrix and generation of communication keys of its own cluster. Also, it is responsible for refreshment of its cluster's data keys. On request, the cluster head nodes generates administrative key of other clusters. In addition to that, the cluster head node is responsible for detecting and evicting compromised sensor nodes in its cluster. Every node is authenticated by the command node right after the initial deployment. After the initial deployment, gateways form their EBS matrices first. Each EBS matrix, along with the list of sensors

in that cluster, is shared between the gateways and the command node. For each cluster, more than one neighbouring cluster head nodes are designated by the command node for managing the administrative keys. For example, if there are 12 keys used in a cluster, command node can designate 4 neighbouring cluster head nodes to manage 3 keys each for that cluster. The cluster head node shares the relevant portions of EBS matrix with each of the neighbouring cluster head node.

Command node shares the key KS_{CH} of each sensor node with its cluster head. It also shares key KS_{Key} of every sensor node with the relevant neighbouring cluster head nodes i.e. the one's responsible for managing any of the administrative keys that will be known to the sensor node. For key distribution, each relevant neighbouring gateway generates one message per individual administrative key in its cluster for each sensor node. The message is first encrypted with the KS_{Key} of the node and then the administrative key of the sensor node's gateway. Gateway decrypts the message, encrypts it with KS_{CH} of the node and sends it to the sensor node. In order to share communication keys, cluster head nodes generate them and send them to their neighbouring cluster head nodes. Neighbouring clusters then send them to sensor nodes in the same way as they send the administrative keys.

If a cluster head node is compromised, either a new cluster head node is deployed or its sensors are redistributed among other cluster head nodes. The new gateway makes a new EBS matrix and repeats the process of initial deployment and initial key distribution. If a sensor node is compromised, keys known to the compromised node are changed with the method mentioned above in the description of EBS matrices. Advantages of SHELL are that it is highly scalable and resilient against node capture attacks. Also, it has very effective node authentication mechanisms. However, it is susceptible to collusion attacks. Collusion attack takes place when two or more compromised nodes

collaborate with each other to attack a sensor network. In the same paper, they have also proposed mechanisms to prevent compromised nodes from collusion by assigning the keys strategically.

LEAP+: LEAP+ [47] is also a state-of-the-art solution for key management in wireless sensor networks. Its initial version was proposed as LEAP [87]. Later on, it was proposed as LEAP+ with some extensions. Although it can be used in both homogeneous and heterogeneous (clustered) sensor networks, it is more suitable for homogeneous sensor networks. This scheme is highly scalable and resistant to collusion attacks. Also, compromised node revocation is very simple and sensor node compromise does not affect other parts of the network.

In LEAP+, every sensor node uses a pseudo-random function to compute keys. Pseudo-random function uses node identities and some pre-loaded key values to compute keys. When sensor nodes are deployed, they compute their individual keys, which they share with only the command node. After that, they exchange their identities with their neighbouring nodes and compute pair-wise keys with their neighbours. In order to broadcast some message, they need a key that is known to all the neighbouring nodes. They compute this key for broadcast purposes and send it to all the neighbouring nodes individually. Lastly, a global key, which is managed by the command node, is used for broadcast in the whole network.

If a sensor node is compromised, all of its neighbouring nodes delete pair-wise keys shared with the compromised node. After that, every neighbouring node computes new value of its key, which is used for broadcast purposes, and sends it to rest of the neighbours individually. Global key is also refreshed in the end. Apart from the increased computation overhead of LEAP+, another drawback of this scheme is that it assumes the network is safe during some initial time period. LEAP+ also has effective

mechanisms for authenticated broadcast.

In wireless sensor networks, it is important to have effective procedures for key refreshment and compromised node revocation. These procedures are not provided by all key management schemes as shown in table 2.2. Trusted KDC provides all three features but it is not scalable due to bottleneck links near the trusted KDC. Also, public key cryptography with authentication provides all three features but it has huge computation costs. Also, it assumes that all sensor nodes can perform complex mathematical computations. Therefore, most appropriate key management schemes, in current literature, are SHELL and LEAP+.

2.5 State-of-the-art Research in WBAN

The use of WBAN in applications, which are crucial for human life, highlights importance of its security. Apart from making sure that a person's biometric information is not tampered with, it is important to ensure confidentiality of the person's information. Key management plays a pivotal role in ensuring data integrity and protecting patient's private data from eavesdroppers and unauthorized users.

WBAN is also a network of sensors like WSN. Therefore, most of the related work in WBAN domain is from WSN domain and many key management schemes, designed for WSN, are applicable to WBAN too. Key management scheme SHELL [49] is not applicable to WBAN because it requires services of neighbouring cluster head nodes, which may not be present in WBAN. Many other schemes are not applicable to WBAN domain because they are designed for large-scale wireless sensor networks. Table **??** shows the applicability of scheme for WSN in WBAN domain.

Apart from that, WBAN differ from WSN in scale, topology, application characteris-

Table 2.2: Comparison of Services Provided by Existing Key Management Schemes for WSN

Scheme	Basic Protection	Key Refreshment	Node Eviction
Single Network-wide Key	Yes	No	No
Pair-wise Key Establishment	Yes	No	No
Random Pair-wise Key Establishment	Yes	No	No
Trusted Key Distribution Center (KDC)	Yes	Yes	Yes
Random Key Pre-distribution	Yes	No	No
Q-Composite Random Key Pre-distribution	Yes	No	No
Multi-path Key Reinforcement	Yes	Yes	No
Polynomial Pool-based Key Pre-distribution	Yes	No	Yes
Grid-based Key Pre-distribution	Yes	No	No
Public Key Cryptography with authentication	Yes	Yes	Yes
SHELL	Yes	Yes	Yes
LEAP+	Yes	Yes	Yes

tics and application environments. For example, if WBAN are employed in hospitals for monitoring serious patients, human intervention will always be possible. Compromised nodes need not be revoked through software under such circumstances because one can conserve energy consumed in node revocation procedure.

For WBAN, researchers have focused on using biometrics as keys and for authentication purposes [34],[35],[36]. Advantage of using biometrics for key computation is that it reduces computations costs associated with random number generation, which is otherwise required for key generation. Also, some researchers have focused on eradicating the need for key exchange [37],[38],[39] assuming that two communicating nodes can sense same biometric at the same time and then apply error-correcting codes to agree on a secret key. Eradicating the need for key exchange eradicates communication costs involved in key management. Apart from time synchronization issues, these schemes add another constraint to the network: they require some sensor nodes to sense more than one biometric. Having multiple sensors in a sensor node increases the cost of sensor node and may not be practical in many WBAN scenarios. Authors in [48] have eradicated time synchronization issues by using photoplethysmogram (PPG) signals for key exchange. To study its efficiency, they have also implemented their scheme in hardware [88]. However, issue of multiple sensing still remains a challenge.

In the discussion up till now, it is evident that applicability of any key management scheme in WBAN is different from its applicability in other classes of sensor networks. This is mainly because of the topology and scale of WBAN. From topology and scale, WBAN resembles WPAN. However, WBAN are used to measure biometrics from human body, which has an effect on communication between sensor nodes planted on human body [24],[25]. Out of the related work mentioned above, I will discuss two important key management schemes in detail as follows: -

Plug 'n Play Key Management for WBAN: [39] proposed a solution for key management in WBAN based on the above mentioned research and studies. They proposed that the communicating sensor nodes do not even need to exchange keys in order to establish a communication link. In this scheme, two sensor nodes sense the same biometric at a particular time instant and then use error correcting codes to compute final key values. Error correcting codes remove the possible differences that may arise in the readings of the two nodes.

This key management scheme is specifically designed for WBAN and is not applicable to generic applications wireless sensor networks. Although it is designed for specifically for WBAN, it is a primitive scheme and has many shortcomings.

Photoplethysmogram (PPG) based key management for WBAN: Most key management schemes for WBAN assume that sensor nodes have perfect time synchronization, communicating sensor nodes can sense same biometric at the same time and then they can communicate using the secret biometric value. This introduces time synchronization issue in WBAN. [48] uses photoplethysmogram values obtained from human body to exchange secret keys between two sensor nodes placed on the same body. [48] uses method devised in [89] to get rid of time synchronization and error-correcting issues.

In photoplethysmogram-based key exchange scheme for WBAN two nodes, who want to share a common secret key, sense plethysmogram values from human body for a certain time period. Then they use the method used in fuzzy vault scheme [89] to agree on a secret value. One of the sensor nodes generate a key value and send it to the other nodes using the secret value, on which both nodes agreed. Although this scheme eradicates the need for time synchronization, it still requires sensor nodes to sense more than one biometric because all sensor nodes must be able to sense photoplethysmogram

in this scheme.

Key management schemes, designed for WSN domain, can be applied to WBAN domain. However, their designs are not efficient when they are applied in WBAN domain. On the other hand, existing key management schemes, designed for WBAN, take assumptions that put extra constraints on sensor nodes. These additional constraints are not practical in many WBAN scenario. Also, key management schemes for WBAN do not take into account node compromise.

2.6 Summary

In this chapter, security threats in wireless sensor networks were listed. Then, existing key management schemes were discussed in detail. Also, they were critically analyzed. Existing key management schemes do provide scalability, energy-efficiency and distributed mechanism to avoid single point of failure. However, these features are not present together in one key management scheme. For WBAN domain, there is no key management scheme that does not put extra constraints on the design of sensor nodes.

Chapter 3

Key Management for WSN

3.1 Introduction

In WSN, group communications are performed to increase efficiency. Groups of nodes share common secret keys. If a node is compromised, it must be evicted from the group and keys must be refreshed in such a way that the compromised node does not get to know the new key values. A single key can not be used for the whole network because in that case, even if a single node is compromised, it compromises the whole network with it. On the other extreme, all pairs of sensor node can have a separate key. This provides high security but it hampers in network processing [52],[69] because some schemes use passive participation of nodes i.e. they decide their actions after overhearing the messages [66],[71]. So, a lightweight scheme, which also enables sharing of a key with large number of nodes, is needed.

Static key management is the simplest form of key management in WSN. It is sometimes also referred to as key pre-distribution, in which keys are calculated and pre-loaded in the nodes before the deployment of the WSN [90],[74],[91],[75],[92],[93],[94]. Intensive research has been done in devising efficient methods for distributing keys before the network deployment [95],[96],[97],[98]. In [99], nodes are pre-loaded with some information, which is used for key establishment after deployment. Camtepe et. al. [100]

have proposed key distribution approach based on combinatorial design using Balanced Incomplete Block Design (BIBD) and Generalized Quadrangles (GQ). These are static key management schemes and they work on the assumption that WSN are very short-lived networks. However, a real-life example of WSN Mica2 has a life-time of two weeks at full power [52]. If keys are not refreshed periodically, there are always chances of cryptanalytic attacks on the WSN.

Many dynamic key management schemes have been proposed, which emphasize on refreshment and revocation of keys periodically. Riaz et. al. [83] proposed a scheme, which actively involves the base station for communication among sensor nodes using public keys. Drawback of this scheme is the frequent communication between sensor nodes and the base station as it incurs a lot of communication overhead. G. Dini [101] proposed a tree-based key revocation protocol for WSNs based on key-chains. Apart from increased storage overhead, another drawback of their scheme is that there is a lot of communication and computation overhead in case of node compromise.

LEAP+ [47] and SHELL [49] are two state-of-the-art schemes for key management in WSN. Also, K.J. Paek et. al. [102] proposed key management based on regional and virtual groups. Drawbacks of Paek et. al. [102] and LEAP+[47] are that they assume the network is safe during some initial time period. Also, all the nodes have to generate keys, which consume a lot of energy. SHELL [49] does not require all the nodes to generate keys, but it has a lot of inter-cluster communication, which is also not desirable. Later, Eltoweissy et. al briefly proposed LOCK [50], which eliminates inter-cluster communication by distributing key generation responsibilities among few nodes within the cluster. However, LOCK requires some nodes to have more capabilities than normal sensor nodes so that they can generate keys. Otherwise, it causes the key generating nodes to die down more quickly.

In this chapter, I propose MUQAMI+, which is a scalable and locally distributed key management scheme for clustered wireless sensor networks. By distributing key management task locally among few sensor nodes, avoid single point of failure in a network or a cluster and increase energy-efficiency key management scheme for WSN. In MUQAMI+, large number of nodes in a cluster share common keys. MUQAMI+ is efficient not only for periodic key refreshment but also for revocation of a compromised node. Moreover, my scheme is flexible and allows role of being cluster head to be transferred from one node to another. Also, my scheme allows the key management responsibilities to rotate among different nodes within the cluster. In addition to that, proposed scheme assumes no initial safe time period. MUQAMI+ is based on Exclusion Basis System (EBS) matrix [84] and key-chains [101]. The key-chain is an authentication mechanism based on Lamport's one-time passwords [103].

Rest of the chapter is organized as follows. Section 3.2 presents the proposed scheme. Section 3.3 and Section 3.4 contain theoretical and simulation based analysis and evaluation respectively. Section 3.5 provides the summary of this chapter.

3.2 MUQAMI+

In my scheme, command node (CN) stores all node IDs. Since the CN does not have energy constraints, I have tried to move as much load to the CN as possible. CN is responsible for managing basic keys (K_{bsc}) and discovery keys (K_{disc}) for all the nodes. It is also responsible for managing keys between Cluster heads and the command node. In order to facilitate in network processing and reduce the overall security overhead, my scheme secures all communications using K_{comm}, which is a group key used for providing group confidentiality. However, using only K_{comm} is very risky and also it can not secure a network against insider attacks. Therefore, I use administrative keys

K_{admin} to secure K_{comm} and to protect the network against insider attacks. Apart from that, every node in the network shares a pair-wise key with its CH. If some key, other than K_{comm}, is required to secure communication between a pair of sensor nodes, the CH node sends a pair-wise key to that pair of sensor nodes directly. In this chapter, I use different abbreviations and notations that are mentioned in Table 3.1.

My scheme uses EBS system of matrices [84] to manage keys. In EBS, a small number of keys are required to manage a large number of nodes. Every node knows a distinct set of k keys out of a set of $k + m$ keys. I have proposed a little change in the representation of EBS matrix. Usually, a '0' is used if a node does not know a key and '1' if a node knows a key. In my scheme, I also use '2', which mean that a node generates a key. Table 3.2 shows example of an EBS matrix. Over three thousand key combinations are available if fourteen keys are used.

After the CH nodes are deployed in the initial phase, the CN sends K_{disc} of all the nodes to their respective CH nodes, so that the CH nodes can recognize the newly deployed nodes in their respective clusters. After the nodes are deployed, the CN computes the details of EBS matrices according to the locations of all the nodes in the network and shares them with the respective CH nodes. CN also sends initial values of relevant administrative keys to each node. While forwarding the encrypted administrative keys to the respective nodes in its cluster, CH nodes also establish pairwise keys with all nodes in its cluster. For key refreshment, CH asks the KG nodes to send new keys to sensor nodes in its cluster. KG nodes compute key values with the help of lightweight one-way hashing functions [104] and broadcast them in the cluster. Keys, other than the administrative and communication keys are very rarely used. Figure 3.1 elaborates the working of my scheme MUQAMI+ with the help of a flow diagram.

Table 3.1: List of Notations Used in Chapter 3

CN	**Command Node or the Base Station**
CH^i	**Cluster Head Node i**
KG^i	**Key Generating Node i. KG nodes compute keys using lightweight one-way hash functions, rather than generating them.**
SN^i	**Sensor Node i**
$\{CH\}$	**Set of all the CH nodes**
$\{SN_{CH^i}\}$	**Set of the SN Nodes belonging to CH i**
$\{KG_{CH^i}\}$	**Set of the KG Nodes belonging to CH i**
K^i_{bsc}	**Basic Key of Node i. Used for communication with the command node. It is preloaded in every node of the network and refreshed after being used once.**
K^i_{disc}	**Discovery Key of Node i. Used for initial discovery of the node. It is preloaded in every node of the network and refreshed after being used once.**
$K^{i,j}_{ch,kg}$	**Key used for communication between CH i and KG j.**
$K^{i,j}_{ch,sn}$	**Key used for communication between CH i and SN j.**
K_{comm}	**Communication Key**
K^i_{admin}	**Administrative Key i**
$K^i_{cn,ch}$	**Key used for communication between CN and CH node i.**
mi	**Message number i in a particular communication sequence.**
$E_K\{A\|B\}$	**Values A and B is put together in a block/chunk and then the chunk is encrypted using Key K**

Table 3.2: Example of an EBS matrix for MUQAMI+ key management scheme for WSN

	N_0	N_1	N_2	N_3	N_4	N_5	N_6	N_7	N_8	N_9
K_1	**2**	**1**	**1**	**1**	**1**	**1**	**0**	**0**	**0**	**0**
K_2	**1**	**2**	**1**	**0**	**0**	**0**	**1**	**1**	**1**	**0**
K_3	**1**	**0**	**0**	**2**	**1**	**0**	**1**	**1**	**0**	**1**
K_4	**0**	**1**	**0**	**1**	**0**	**2**	**1**	**0**	**1**	**1**
K_5	**0**	**0**	**1**	**0**	**2**	**1**	**0**	**1**	**1**	**1**

3.2.1 Initial Deployment

CH nodes are deployed in the first phase. Following is the first message that a newly deployed CH i sends to the CN: -

$$m1 : \forall CH^i \in \{CH\} : CH^i \rightarrow CN : E_{K^i_{disc}}\{ID|Auth\_Code\}$$

Then for every CH i, the CN authenticates it and sends to it the $K^i_{cn,ch}$ and the EBS matrix of its cluster along with IDs and K_{disc} of all nodes j, which are to be deployed in the cluster of CH i: -

$$m2 : \forall CH^i \in \{CH\} : CN \rightarrow CH^i : E_{K^i_{disc}}\{K^i_{cn,ch}|EBS\_Matrix$$

$$|\forall SN^j \in \{\{SN_{CH^i}\} \cup \{KG_{CH^i}\}\} : \{ID(SN^j)|K^j_{disc}\}\}\}$$

In the above message, IDs and K_{disc} of all the relevant nodes are put together in a block along with the $K^i_{cn,ch}$ and the relevant EBS matrix, then encrypted using K_{disc} of the CH node and then sent to the CH i from the CN. SN and KG nodes are deployed in the second phase. Following messages are exchanged for every KG node j that is deployed

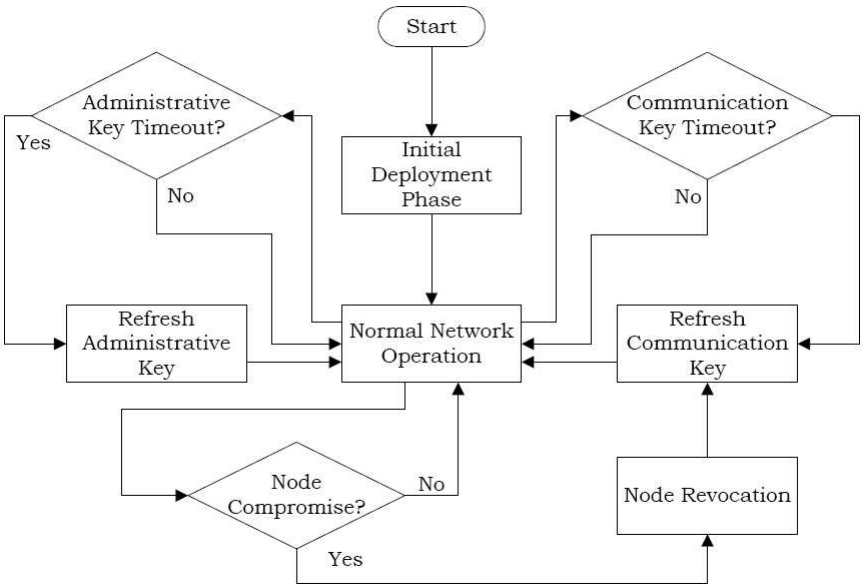


Figure 3.1: Outline of the proposed scheme MUQAMI+ for WSN

in the cluster i: -

$$\forall CH^i \in \{CH\} \wedge \forall KG^j \in \{KG_{CH^i}\} :$$

$$m1 : KG^j \rightarrow CH^i : E_{K^j_{disc}}\{ID(KG^j)|Auth\_Code\}$$

$$m2 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{ID(KG^j)|Auth\_Code\}$$

$$m3 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{E_{K^j_{bsc}}\{K^j_{bsc\_new}$$
$$|K^j_{disc\_new}|K^1_{admin}|K^2_{admin}|...|K^{k-1}_{admin}\}\}$$

$$m4 : CH^i \rightarrow KG^j : E_{K^j_{disc}}\{K^{i,j}_{ch,kg}|E_{K^j_{bsc}}$$
$$\{K^j_{bsc\_new}|K^j_{disc\_new}|K^1_{admin}|K^2_{admin}|...|K^{k-1}_{admin}\}\}$$

In the above messages, after authenticating a new KG node j i.e. after the first two messages, CN puts together all the K_{admin} relevant to the KG node j in a block along

with the new values of K_{bsc} and K_{disc}, encrypts this block first using the current value of K_{bsc} and then using $K^{i}_{cn,ch}$ and then sends it to the CH i in message $m3$. After receiving $m3$, CH i generates the seed value for $K^{i,j}_{ch,kg}$ and computes the whole key chain associated with $K^{i,j}_{ch,kg}$. CH i then adds the seed value of $K^{i,j}_{ch,kg}$ into the block and sends it to KG j in $m4$. Note that the CN sends new values for K_{bsc} and K_{disc} every time they are used. Also, k-1 administrative keys are communicated to a KG node as it is responsible for generating one key by itself. After receiving $m4$, KG j computes the associated key-chain for $K^{i,j}_{ch,kg}$. Similar message exchanges take place for every SN node j that is deployed in cluster i: -

$$\forall CH^i \in \{CH\} \wedge \forall SN^j \in \{SN_{CH^i}\} :$$

$$m1 : SN^j \rightarrow CH^i : E_{K^j_{disc}}\{ID(SN^j)|Auth\_Code\}$$

$$m2 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{ID(SN^j)|Auth\_Code\}$$

$$m3 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{E_{K^j_{bsc}}\{K^j_{bsc\_new}$$
$$|K^j_{disc\_new}|K^1_{admin}|K^2_{admin}|...|K^k_{admin}\}\}$$

$$m4 : CH^i \rightarrow SN^j : E_{K^j_{disc}}\{K^{i,j}_{ch,sn}|E_{K^j_{bsc}}$$
$$\{K^j_{bsc\_new}|K^j_{disc\_new}|K^1_{admin}|K^2_{admin}|...|K^k_{admin}\}\}$$

No key-chain is associated with $K^{i,j}_{ch,sn}$ as it is rarely used. Sometimes, a node is not deployed in its expected cluster. In that case, messages $m2$ and $m3$ in the above message exchanges will be changed as follows: -

$$m2 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{E_{K^j_{disc}}\{ID(SN^j)|Auth\_Code\}\}$$

$$m3 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{K^j_{disc}|E_{K^j_{bsc}}\{K^j_{bsc\_new}$$
$$|K^j_{disc\_new}|K^1_{admin}|K^2_{admin}|...|K^k_{admin}\}\}$$

In the end, CN shares final version of the EBS matrix with the CH node as follows: -

$$m1 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{EBS\_Matrix\}$$

Note that the cluster heads do not know the administrative keys being used in their clusters. This is important to avoid single point of failure i.e. revelation of all K_{admin} in case of compromise of CH. Next, the initial values of communication keys are distributed. Every CH node i sends communication key to all KG nodes in its cluster. In turn, every KG node broadcasts the communication key in the cluster using the administrative keys that it manages. The message exchanges for broadcasting the initial values of communication keys are as follows: -

$$\forall CH^i \in \{CH\} \wedge \forall KG^j \in \{KG_{CH^i}\} :$$
$$m1 : CH^i \rightarrow KG^j : E_{K^{i,j}_{ch,kg}}\{K^i_{comm}\}$$
$$m2 : KG^j \rightarrow * : E_{K^j_{admin}}\{K^i_{comm}\}$$

3.2.2 Re-keying and Node addition

In order to avoid the cryptanalytic attacks on the network, keys need to be refreshed regularly. Communication keys are refreshed in the same manner as they were distributed initially i.e. using the administrative keys. Administrative keys are refreshed using their previous values. In order to refresh K^l_{admin} of its own cluster, CH node i sends a refresh message to the KG node j, which manages K^l_{admin}. KG node j then broadcasts the new administrative key encrypted in the old one. Following message exchanges take place: -

$$m1 : CH^i \rightarrow KG^j : E_{K^{i,j}_{ch,kg}}\{Refresh\_Message\}$$
$$m2 : KG^j \rightarrow * : E_{K^l_{admin}}\{K^l_{admin\_new}\}$$

When a sensor node receives new value of an administrative key, it verifies the new value through the one-way hashing function as follows: -

$$K^l_{admin} = F(K^l_{admin\_new})$$

where F is the one-way hashing function that is used to compute the administrative keys. Since key-chains are used to manage $K_{ch,kg}$ and K_{admin}, it becomes necessary for the KG node to get the new seed value from CH node or the CN node respectively. A KG node j gets the new value of K^l_{admin}, which it manages, from the CN through its CH i as follows: -

$$m1 : KG^j \rightarrow CH^i : E_{K^{i,j}_{ch,kg}}\{E_{K^j_{bsc}}\{Auth\_Code|Refresh\_Msg\}\}$$

$$m2 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{E_{K^j_{bsc}}\{Auth\_Code|Refresh\_Msg\}\}$$

$$m3 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^l_{admin\_seed}\}\}$$

$$m4 : CH^i \rightarrow KG^j : E_{K^{i,j}_{ch,kg}}\{E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^l_{admin\_seed}\}\}$$

KG node j can get the new seed value for $K^{i,j}_{ch,kg}$ from the CH node i using the last value of $K^{i,j}_{ch,kg}$ in the key-chain.

For the addition of SN node j in cluster i, CN sends the following message to the CH i: -

$$m1 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{ID(SN^j)|K^j_{disc}\}$$

After getting this message, CH i waits for the discovery message from the SN node j. When deployed, SN node j will contact the CH i with its discovery key K^j_{disc}. Following

message exchanges will take place to add the new SN node j in cluster i: -

$$m2 : SN^j \rightarrow CH^i : E_{K^j_{disc}}\{ID(SN^j)|Auth\_Code\}$$

$$m3 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{ID(SN^j)|Auth\_Code|Cur\_Admin\_Chain\_Indexes\}$$

$$m4 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^j_{disc\_new}|K^1_{admin}|K^2_{admin}$$
$$|...|K^k_{admin}\}\}$$

$$m5 : CH^i \rightarrow SN^j : E_{K^j_{disc}}\{K^{i,j}_{ch,sn}|E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^j_{disc\_new}|K^1_{admin}$$
$$|K^2_{admin}|...|K^k_{admin}\}\}$$

where $Cur\_Admin\_Chain\_Indexes$ represents the number of times K_{admin}, related to SN node j, has been refreshed. Based on $Cur\_Admin\_Chain\_Indexes$, the CN calculates and sends to the SN j the current values of the K_{admin} related to the SN node j. In case a new KG node j is to be deployed in cluster i, CN sends initial value of the new key 'l', which KG j manages, to all the relevant SN nodes in the cluster through the CH i. Then the new KG node j is deployed in the same manner in which a new SN node is deployed. Only difference is that $k-1$ admin keys are sent to the new KG node as it generates one by itself. The fact that the CN is often solicited through CH nodes has an impact on the energy consumption of CH nodes. A single node may not be able to act as a CH node throughout the network lifetime. Therefore, my scheme has the flexibility to shift responsibility of being CH node from the current CH node to another node, which has the capability of becoming CH. Refer to Section 3.2.3.1 for details regarding the addition a new CH node.

3.2.3 Node Compromise

If a node is compromised, one needs to refresh keys in such a way that the new keys are not known to the compromised node and it can only act as an outsider when trying to interfere in the network operation. I assume that an efficient mechanism to detect an attack is already in place and the relevant CH node starts the recovery procedure. In case of CH node compromise, CN starts the procedure. There are three types of node in MUQAMI+. I will discuss the implications of the compromise of each type of node one by one.

3.2.3.1 Cluster Head Compromise

If a CH node is compromised, CN can either deploy a new CH node or designate an existing node from the network to act as a CH node. Apart from sharing the discovery keys of all nodes in the cluster and the EBS matrix of the cluster i with the new CH i, CN sends a validation message to each node in the cluster through the new CH node i. CH i can not decrypt the validation messages as they are encrypted using K_{bsc} of the related nodes. Following message is exchanged between CN and the new CH node i: -

$$m1 : CN \rightarrow CH^i : E_{K^i_{bsc}}\{EBS|\forall l \in \{\{SN_{CH^i}\} \cup \{KG_{CH^i}\}\} :$$
$$\{K^l_{disc}|E_{K^l_{bsc}}\{K^l_{bsc\_new}|K^l_{disc\_new}|CH\_Valid\}\}\}$$

Then the CH node sends the validation messages to all the SN nodes k along with the new value of $K^{i,k}_{ch,sn}$. For all the KG nodes j, it will send the validation message along with a new seed value of $K^{i,j}_{ch,kg}$. The new CH node i will send the following messages

to each each SN node k and KG node j respectively: -

$$m2 : \forall SN^k \in \{SN_{CH^i}\} : CH^i \rightarrow SN^k : E_{K^k_{disc}}$$
$$\{K^{i,k}_{ch,sn} | E_{K^k_{bsc}} \{K^k_{bsc\_new} | K^k_{disc\_new} | CH\_Valid\}\}$$
$$m2 : \forall KG^j \in \{KG_{CH^i}\} : CH^i \rightarrow KG^j : E_{K^j_{disc}}$$
$$\{K^{i,j}_{ch,kg} | E_{K^j_{bsc}} \{K^j_{bsc\_new} | K^j_{disc\_new} | CH\_Valid\}\}$$

3.2.3.2 Sensor Node Compromise

If an SN node is compromised, one needs to distribute the set of K keys known to the compromised SN node using the M keys not known to the compromised SN node. So, in the first phase, the CH node will ask all the KG nodes, which generate those K keys, to generate new values, encrypt them using the previous ones and send them back to the CH node. If the SN node is compromised in cluster i, following communications will take place between CH i and the KG nodes, which manage those K keys in the cluster: -

$$m1 : \forall p \in K : CH^i \rightarrow KG^p : E_{K^{i,p}_{ch,kg}} \{Revoc\_Msg\}$$
$$m2 : \forall p \in K : KG^p \rightarrow CH^i : E_{K^{i,p}_{ch,kg}} \{E_{K^p_{admin}} \{K^p_{admin\_new}\}\}$$

Now the CH node will aggregate these K encrypted values and send the aggregated message to the M KG nodes i.e. those KG nodes, which manage keys that are not known to the compromised node. Each one of those M KG nodes will then broadcast the aggregated messages using the key it manages. SN nodes, which uses any of those K compromised keys, will get the new value using some key that it knows other than those K keys. Note that no two nodes know the same set of K keys in the EBS matrix (Refer to Table 3.2). Following message exchanges will take place to distribute the refreshed

keys: -

$$m3 : \forall q \in M : CH^i \rightarrow KG^q : E_{K^{i,q}_{ch,kg}}\{\forall p \in K : E_{K^p_{admin}}\{K^p_{admin\_new}\}\}$$

$$m4 : \forall q \in M : KG^q \rightarrow * : E_{K^q_{admin}}\{\forall p \in K : E_{K^p_{admin}}\{K^p_{admin\_new}\}\}$$

Note that in all these communications, the compromised SN node can not use the keys known to it in order to interfere in the network operations.

3.2.3.3 Key-Generator Compromise

If a KG node is compromised, either a new node will be deployed or an existing node will be given the responsibility of managing the key, which was previously managed by the compromised node. If a new node is deployed, key-chain will be pre-loaded into it before deployment. It will only need to know the current key value so that it can use it to send the new value. On the other hand, if an existing node is given the responsibility, it will also need a seed value to compute the new key-chain. In order to award the responsibility to an existing SN j in cluster i, following messages are exchanged: -

$$m1 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{Revoc\_Msg|Key\_ID|Cur\_Refr\_Iter\}$$

$$m2 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{ID(SN^j)|K^j_{disc}|E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^j_{disc\_new}$$

$$|Cur\_Key\_Val|Seed\_Val\}\}$$

$$m3 : CH^i \rightarrow SN^j : E_{K^j_{disc}}\{E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^j_{disc\_new}|Cur\_Key\_Val|Seed\_Val\}\}$$

where $Seed\_Val$ is the seed value of the K_{admin} that SN^j has to manage. After these messages are exchanged, SN^j becomes one of the K KG nodes, which know one compromised key each in the cluster i. Similar procedure, as in the previous section (Section 3.2.3.2), is followed to distribute the set of K compromised keys using the remaining set of M keys.

Since a compromised node is not bound in its behaviour, it may happen that it refreshes a key without consent of the CH node. If the compromised KG node has already refreshed the compromised key without the instructions of the CH i, then new initial value of the compromised key, encrypted with the respective K_{bsc} of all the relevant SN nodes, is sent to all the relevant SN nodes through the CH node i. In such scenario, following message exchanges will take place instead of the above message exchanges: -

$$m1 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{Revoc\_Msg|Key\_ID\}$$

$$m2 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{ID(SN^j)|K^j_{disc}|E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^j_{disc\_new}$$
$$|Seed\_Val\}|\forall SN^l \in Key\_ID : ID(SN^l)|K^l_{disc}|E_{K^l_{bsc}}\{K^l_{bsc\_new}|K^l_{disc\_new}$$
$$|New\_Init\_Key\_Val\}\}$$

$$m3 : CH^i \rightarrow SN^j : E_{K^j_{disc}}\{E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^j_{disc\_new}|Seed\_Val\}\}$$

$$m4 : \forall SN^l \in Key\_ID : CH^i \rightarrow SN^l : E_{K^l_{disc}}\{E_{K^l_{bsc}}\{K^l_{bsc\_new}|K^l_{disc\_new}$$
$$|New\_Init\_Key\_Val\}\}$$

3.3 Analysis and Comparison

In this section, I will provide a brief comparison of MUQAMI+ with other schemes and try to establish my claims. Two factors contribute towards the usage of power in a node: communication overhead and computation overhead. However, one needs to consider the storage overhead first as the sensor nodes are also limited in their storage capacity.

3.3.1 Storage Overhead

EBS based key management schemes are inherently able to support a large number of nodes with a small number of keys using combinatorics (See Figure 3.2). Note that the

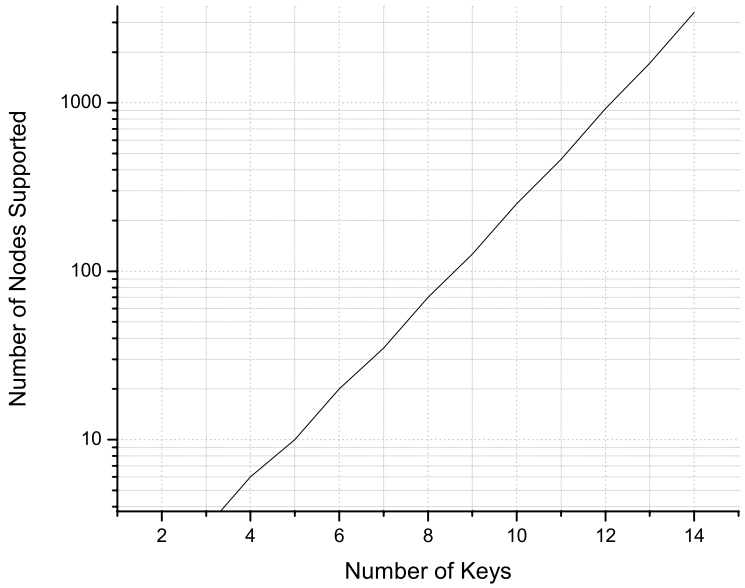


Figure 3.2: Number of nodes that can be supported using EBS matrix for key management

graph is drawn on logarithmic scale because the number of nodes that can be supported increases exponentially with respect to the number of keys used. This is particularly helpful when compromised nodes need to be revoked. Number of nodes n that can be supported using EBS matrix is given by the formula: -

$$n = \frac{(k+m)!}{k!m!} \tag{3.1}$$

where k and m are EBS parameters. As opposed to SHELL [49], cluster heads in my scheme need not store any key to communicate with other cluster heads. Moreover, gateways in my scheme also need not generate and store EBS keys for other clusters. In MUQAMI+, each CH node has to store one K_{comm} and $K_{cn,ch}$ apart from $K_{ch,sn}$ of all SN nodes and the key-chains $K_{ch,kg}$ of all KG nodes in the cluster. So, the average storage requirement of a CH node (in number of keys) in MUQAMI+ can be expressed

with the following formula: -

$$SR_{CH}^{MUQAMI+} = (l \times (k+m)) + r - (k+m) + 2 \tag{3.2}$$

where l is length of key chain and r is the number of nodes in a cluster. SN nodes have to store k admin keys apart from four other keys: $K_{ch,sn}$, K_{comm}, K_{bsc} and K_{disc}. So, the average storage requirement of an SN node in MUQAMI+ can be expressed with the following formula: -

$$SR_{SN}^{MUQAMI+} = k + 4 \tag{3.3}$$

KG nodes have to store two key-chains: one for the admin key, which it generates and one for $K_{ch,kg}$. Also, it has to store $k-1$ EBS keys along with three other keys: K_{comm},K_{bsc} and K_{disc}. So the storage requirement of a KG node can be expressed as: -

$$\begin{aligned} SR_{KG}^{MUQAMI+} &= 2l + (k-1) + 3 \\ &= 2(l+1) + k \end{aligned} \tag{3.4}$$

Since there are $k+m$ KG nodes out of r nodes inside the cluster, average storage requirement of each node within a cluster comes out to be: -

$$\begin{aligned} SR_{SN \cup KG}^{MUQAMI+} &= \frac{(r-(k+m))(k+4) + (k+m)(2(l+1)+k)}{r} \\ &= \frac{r(k+4) + (k+m)(2(l+1)-4)}{r} \\ &= (k+4) + \frac{2(l-1)(k+m)}{r} \end{aligned} \tag{3.5}$$

Note that the ratio $((k+m) : r)$ is very small as $(k+m) << r$ (See Equation 3.1 and Figure 3.2). Therefore, average storage requirements of a node inside a cluster is not too much as compared to SHELL.

In my scheme, I use one-way hashing functions to compute key chains. Also, key chain length l is a variable, which can change according to the storage capabilities of

a node. However, l must be a value such that the cost of computing the keys does not exceed the cost of generating them on the nodes. In other words, following inequality must hold: -

$$\begin{aligned} & Cost_{seed} + l(Cost_{comp} - 1) < l(Cost_{gen}) \\ & \Rightarrow Cost_{seed} < l(Cost_{gen} - Cost_{comp} + 1) \\ & \Rightarrow l < \frac{Cost_{seed}}{Cost_{gen} - Cost_{comp} + 1} \end{aligned} \tag{3.6}$$

where $Cost_{seed}$, $Cost_{comp}$ and $Cost_{gen}$ are the costs of getting new seed value, computing a key value through one-way hashing function and generating a key on a node respectively.

Average storage requirements of a CH node in LEAP+ is fairly straightforward. Apart from the pairwise key shared with each node in the cluster, it has to store two more keys i.e. its cluster key and the communication key. so, if there are r nodes in a cluster, storage requirements of a CH node in LEAP+ turns out to be: -

$$SR_{CH}^{LEAP+} = r + 2 \tag{3.7}$$

In LEAP+, the SN nodes only establish pair-wise keys with only their b neighbours. However, they have to store two keys i.e. the cluster key and the communication key. So, the storage requirement of SN node in LEAP+ becomes:-

$$SR_{SN}^{LEAP+} = b + 2 \tag{3.8}$$

Now, I will discuss the storage requirement of a CH node in SHELL. In SHELL, each node has to store a key-chain of length l for the key it shares with the CN. Also, it has to store pair-wise keys with h neighbouring CH nodes, with whom it shares the EBS matrix. Also, it has to store pair-wise keys with the r nodes i.e. average number of nodes

in a cluster. Finally, it has to store the $k+m$ administrative keys and the communication key. so, the storage requirement of a CH node in SHELL can be written as

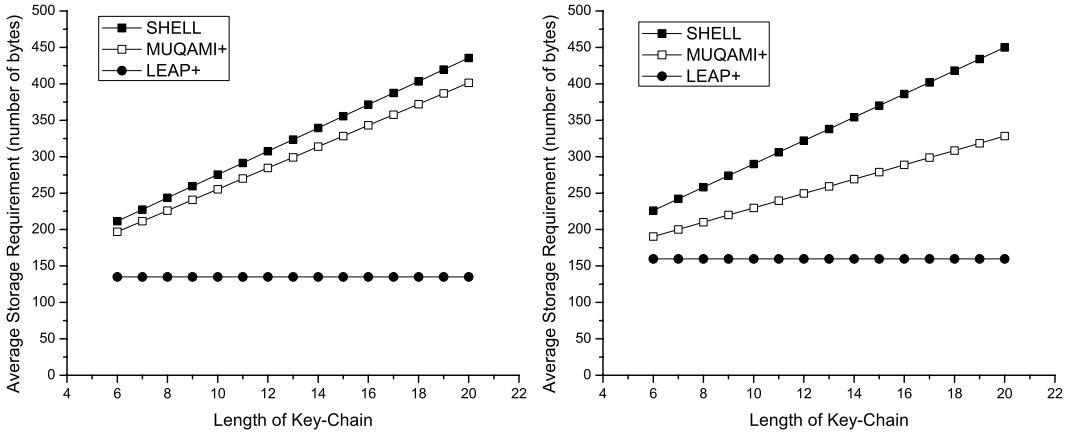
$$SR_{CH}^{SHELL} = l + r + h + k + m + 1 \tag{3.9}$$

Apart from the key-chain of length l for the key shared with the CN and the k administrative keys, SN nodes have to store three other keys i.e. one pair-wise key shared with a neighbouring CH node, one shared with its own CH node and one communication key. So the average storage requirements of a SN node in SHELL can be written as: -

$$SR_{SN}^{SHELL} = l + k + 3 \tag{3.10}$$

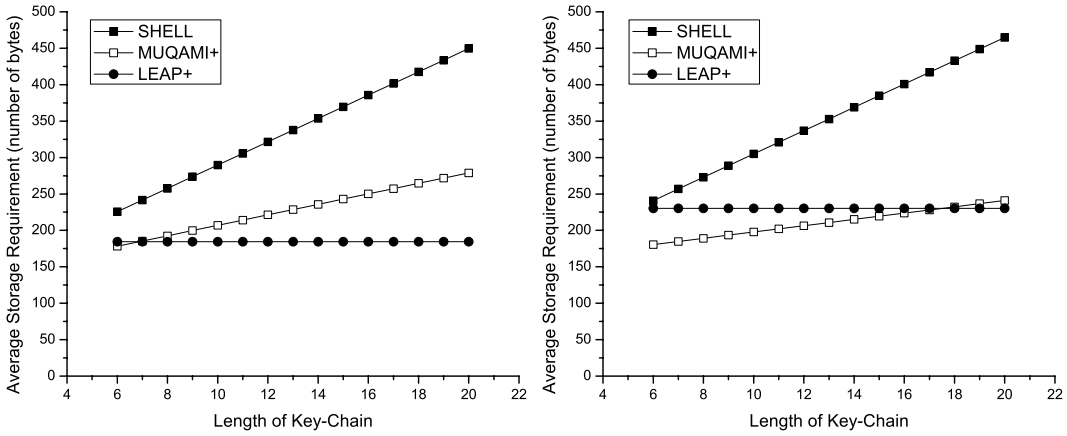
Table 3.3 compares the storage requirements of MUQAMI+ with other schemes. In Table 3.3, b is the average number of neighbouring nodes, with whom SN node has to share pair-wise keys and h is the number of neighbouring CH nodes, with whom a CH node communicates. Value of h can vary depending upon the value of EBS parameters k and m and the extent, to which the distribution of keys is desired. My scheme is completely distributed in nature i.e. one node manages not more than one key. Value of b also varies according to the size and density of network. However, the values of parameters h and b are always such that they are comparable to the value of $k+m$. In Table 3.3, I use SN nodes to represent both SN and KG nodes of MUQAMI+ because other schemes do not have any KG nodes.

One can decrease the key chain length in MUQAMI+ at the cost of more computations and communications but one can not store more keys in LEAP+ to reduce the computation and communication costs. This proves my scheme to be more adaptable to the capabilities of nodes as compared to LEAP+. Also, storage requirements of SN nodes is very low in my scheme MUQAMI+ as compared to SHELL. Figure 3.3 compares memory consumption of our scheme with SHELL and LEAP+ schemes.



(a) Average Memory Required by a Node in a cluster of 22 nodes

(b) Average Memory Required by a Node in a cluster of 38 nodes



(c) Average Memory Required by a Node in a cluster of 59 nodes

(d) Average Memory Required by a Node in a cluster of 110 nodes

Figure 3.3: Comparison of Average Storage Requirement in a Node using SHELL, LEAP+ and MUQAMI+ varying length of key-chain and node density in a cluster (High Node Density means Higher Number of Neighbouring Nodes)

Table 3.3: Storage requirements (in number of keys) of each type of node in SHELL, LEAP+ and MUQAMI+ schemes in WSN

	CH	**SN**
MUQAMI+	$(l \times (k+m)) + r - (k+m) + 2$	$(k+4) + [(2(l-1)(k+m))/r]$
LEAP+	$r + 2$	$b + 2$
SHELL	$l + r + h + k + m + 1$	$l + k + 3$

3.3.2 Communication and Computation Overhead

Since communication is the most energy consuming activity, I will analyze and discuss it in more detail. While designing my scheme, I tried to minimize and localize the communication as much as possible. This also helps in reducing the computation overhead because less message exchanges will result in lesser encryptions/decryptions. Now I compare my scheme with other schemes with respect to the number of message exchanged between various types of nodes during different phases.

In the initial deployment phase of SHELL, every CH node receives one message each from h neighbouring CH nodes to establish the communication path. Also, for each node it its cluster, the CH node receives one messages from one of the neighbouring CH nodes in order to establish pair-wise keys. Finally, for the distribution of the communication key, every CH node receives the communication keys $k + m$ times i.e. encrypted separately in each administrative key. So, the average message exchanges between CH nodes during initial deployment phase of SHELL come out to be: -

$$Avg\_Msg\_Count\_Init^{SHELL}_{CH \rightarrow CH} = h + r + k + m \tag{3.11}$$

where $Avg\_Msg\_Count\_Init^{SHELL}_{CH \rightarrow CH}$ is the average message exchanges between CH

nodes during initial deployment phase of SHELL.

In the initial deployment phase of LEAP+, every SN node sends two unicast messages to the CH node. One message establishes pair-wise key with the CH node and the other one transfers the cluster key to the CH node. Apart from that, it also has to forward d messages to the CH node, where d is the average number of nodes that establish their pair-wise keys with the CH node through each node. So, the average number of messages transmitted from a SN to its CH in the initial deployment phase turn out to be: -

$$Avg\_Msg\_Count\_Init_{SN \rightarrow CH}^{LEAP+} = d + 2 \tag{3.12}$$

where $Avg\_Msg\_Count\_Init_{SN \rightarrow CH}^{LEAP+}$ is the average number of messages transmitted from a SN to its CH in the initial deployment phase of LEAP+. Also, it is important here to mention the average number of message exchanges between SN nodes in LEAP+. There are b neighbours of each node and each node has to send two messages to each of its neighbours. One message is for the pair-wise key establishment and the other message is for the communication of its cluster key. Apart from that, three messages are exchanged for every node that establishes a pair-wise key with the CH node through this node i.e. three messages are exchanged for d nodes from each node on average. In one message, the d nodes send a message to the CH node through this node to establish pair-wise key with the CH node. In the other two messages, cluster keys are exchanged between the CH and the d nodes i.e. CH and the d nodes send their cluster keys to each other through this node. So, the average number of message exchanges between SN nodes in LEAP+ for the initial deployment phase comes out to be: -

$$Avg\_Msg\_Count\_Init_{SN \rightarrow SN}^{LEAP+} = 2b + 3d \tag{3.13}$$

where $Avg\_Msg\_Count\_Init_{SN \rightarrow SN}^{LEAP+}$ is the average number of messages exchanged between SN nodes in the initial deployment phase of LEAP+.

If I assume key length to be x bytes, EBS matrix size to be y bytes and average length of other types of messages to be z bytes, then Table 3.4 shows the average number of messages transmitted by each type of node on each link during the initial deployment phase of each scheme. Here g is the total number of CH nodes in the network and d is the number of nodes, which communicate with the CH node through a particular node. Exact value of d depends upon the network topology and varies from node to node even within a network. One important thing to note in Table 3.4 is that there is very less load on CH nodes in LEAP+. In LEAP+, SN nodes have to bear extra load instead of the CH node in the initial deployment phase. LEAP+ is designed in such a way that if two nodes have to share a key, one of them broadcasts and then the other one unicasts. I could have assumed that the CH node uses the broadcasts of SN nodes. However, that exercise would have only increased the load on CH node without any significant effect on the load of SN nodes. The reason is that every SN node would have to broadcast with even more power so that its broadcast message could reach the CH node. New nodes are added in the same way as they are deployed initially in all the three schemes. So the comparison of new node addition would be similar to the one shown in Table 3.4.

For key refreshment in SHELL, every CH sends a message to h neighbouring CH nodes. In turn, it receives $k + m$ messages from them to broadcast in the cluster. So, the average number of messages exchanged between the CH nodes in key refreshment phase can be written as: -

$$Avg\_Msg\_Count\_Rekey_{CH \rightarrow CH}^{SHELL} = h + k + m \tag{3.14}$$

where $Avg\_Msg\_Count\_Rekey_{CH \rightarrow CH}^{SHELL}$ is the average message exchanges between CH nodes during the key refreshment phase of SHELL. Also, the keys between the neighbouring CH nodes and the keys between CH nodes and the CN will also be refreshed. So, in order to refresh those keys, the CN will send $h + 1$ messages to all the CH nodes

Table 3.4: Average number of **bytes** transmitted by each type of node on each link during initial deployment phase(* means a broadcast within cluster) of SHELL, LEAP+ and MUQAMI+ in WSN

	MUQAMI+	**LEAP+**	**SHELL**
$CH \rightarrow CH$	-	-	$x(h+r+k+m)$
$CH \rightarrow CN$	$2z(r+1)$	-	$2z$
$CN \rightarrow CH$	$g(x+y+r(z+x)+x(k+2))$	$g(2x)$	$g(x+(h+r)(z+x)+z)$
$CH \rightarrow *$	-	$2x$	$x(k+m)$
$CH \rightarrow SN$	$r(k+3)x$	rx	rkx
$SN \rightarrow CH$	$2z$	$(d+2)x$	$2z$
$SN \rightarrow SN$	-	$(2b+3d)x$	-
$SN \rightarrow *$	-	$2x$	-

i.e. to g CH nodes. h messages contain new keys for communication with the neighbouring CH nodes and one message contains key for communication with the command node. So, the total number of messages transferred from CN to the CH nodes in key refreshment phase of SHELL comes out to be: -

$$Msg\_Count\_Rekey_{CN \rightarrow CH}^{SHELL} = g \times (h+1) \tag{3.15}$$

where $Msg\_Count\_Rekey_{CN \rightarrow CH}^{SHELL}$ is the total number of messages transferred from CN to the CH nodes in key refreshment phase of SHELL.

In order to refresh its cluster key in LEAP+, each node sends one message to the CH node. Also, it forwards one message each from d nodes, So, the average number of

messages transmitted from a SN to its CH in the key refreshment phase turns out to be: -

$$Avg\_Msg\_Count\_Rekey_{SN \rightarrow CH}^{LEAP+} = d + 1 \tag{3.16}$$

where $Avg\_Msg\_Count\_Rekey_{SN \rightarrow CH}^{LEAP+}$ is the average number of messages transmitted from a SN to its CH in the key refreshment phase of LEAP+. In the same way, when a SN node refreshes its cluster key in LEAP+, it sends one message each to its b neighbours. Also, the CH node exchanges cluster keys with d nodes through every node on average. On a particular SN node, one message is received from each of the d nodes and forwarded to the CH node and one message from the CH node is forwarded to it. So, the average number of message exchanges between SN nodes in LEAP+ for the key refreshment phase comes out to be: -

$$Avg\_Msg\_Count\_Rekey_{SN \rightarrow SN}^{LEAP+} = b + 2d \tag{3.17}$$

where $Avg\_Msg\_Count\_Rekey_{SN \rightarrow SN}^{LEAP+}$ is the average number of messages exchanged between SN nodes in the key refreshment phase of LEAP+.

Communication keys are refreshed in the same way as the administrative keys except in LEAP+. In LEAP+, communication key refreshment is comparatively efficient for CH nodes. However, this efficiency comes at the cost of increased load on all SN nodes. Average number of messages broadcasted by an SN node for rekeying in my scheme can be expressed with the following formula: -

$$Avg\_Msg\_Count\_Rekey_{SN \rightarrow *}^{MUQAMI+} = \frac{k + m}{r} \tag{3.18}$$

Apart from that, I have added an expression $1/l$ in number of communications from $SN \rightarrow CH$, $CH \rightarrow SN$ and $CH \rightarrow CN$. This expression caters for the communications that are required to get a new seed value from the CH node. Table 3.5 shows the average number of bytes transmitted by each type of node during the key refreshment phase.

Table 3.5: Average number of **bytes** transmitted by each type of node on each link during key refreshment phase(* means a broadcast within cluster) of SHELL, LEAP+ and MUQAMI+ schemes in WSN

	MUQAMI+	**LEAP+**	**SHELL**
$CH \rightarrow CH$	-	-	$x(h+k+m)$
$CN \rightarrow CH$	$(zg)+((2xg)/l)$	xg	$xg(h+1)$
$CH \rightarrow CN$	$(2z)/l$	-	-
$CH \rightarrow *$	-	-	$x(k+m)$
$CH \rightarrow SN$	$x(k+m)+((2z)/l)$	xr	-
$SN \rightarrow CH$	$(2z)/l$	$x(d+1)$	-
$SN \rightarrow SN$	-	$x(b+2d)$	-
$SN \rightarrow *$	$x((k+m)/r)$	-	-

If a CH node is compromised, recovery procedure of my scheme is fairly straight forward as compared to other schemes. In Table 3.6, I have compared the communication overhead of my scheme with other schemes in case of CH node compromise. Expressions in Table 3.6 are similar to the ones in Table 3.4 except that CN sends $h+1$ messages to the CH nodes in case a CH node is compromised. One message is sent to the new CH node and h messages are sent to neighbouring CH nodes, so that keys can be established between the new CH node and its neighbouring CH nodes. Explanations for the rest of the expressions in Table 3.6 are similar to the ones in Table 3.4.

If an SN node is compromised, I do not assume that a new SN node is deployed. However, in case of KG node in MUQAMI+, one has to give the responsibility of key generation to some other node. In MUQAMI+, if a KG node is compromised the CH

Table 3.6: Average number of bytes transmitted by each type of node, in WSN using SHELL, LEAP+ and MUQAMI+, on each link in case the cluster head node of the cluster is compromised. All communications are within the cluster of the compromised cluster head except $CH \rightarrow CH$ communication (* means a broadcast within cluster)

	MUQAMI+	**LEAP+**	**SHELL**
$CH \rightarrow CH$	-	-	$x(h + (k + m)(2r + 1))$
$CN \rightarrow CH$	$y + r(3x + z)$	$2x$	$(2xh) + r(2x + z)$
$CH \rightarrow CN$	$2z$	$2z$	$2z$
$CH \rightarrow *$	-	$2x$	$(k + m)x$
$CH \rightarrow SN$	$(r(3x+z))+x(k+m)+((2z)/l)$	rx	$r((k + 3)x + z)$
$SN \rightarrow CH$	-	$(d + 2)x$	-
$SN \rightarrow SN$	-	$(2b + 3d)x$	-
$SN \rightarrow *$	-	-	-

sends one message to the CN and the CN replies with the compromised Administrative key's new seed value. Apart from that, there is only one extra communication from CH to SN node, after which the SN node becomes a KG node. I assume that all nodes in a cluster have equal probability of being compromised. So, the average number of communications between CH and CN, in case of the compromise of SN or KG node, will be similar to Equation 3.5 i.e. $(k+m)/r$. Similarly, the average number of communications

from CH to SN node can be calculated with the following formula: -

$$\begin{aligned} Avg\_Msg\_Count\_SNCompr^{MUQAMI+}_{CH\rightarrow\{SN\cup KG\}} \\ &= \frac{((r-(k+m))(k+m))+(k+m)(k+m+1)}{r} \\ &= \frac{(k+m)(r-k-m+k+m+1)}{r} \\ &= \frac{(k+m)(r+1)}{r} \end{aligned} \tag{3.19}$$

Whereas the average number of bytes transferred from CH to SN node for the above task can be expressed with the following formula: -

$$\begin{aligned} Avg\_Byte\_Count\_SNCompr^{MUQAMI+}_{CH\rightarrow\{SN\cup KG\}} \\ &= \frac{((r-(k+m))(kz+mx))+(k+m)(kz+mx+2z+2x)}{r} \\ &= \frac{krz+mrz+2zk+2xk+2zm+2xm}{r} \\ &= \frac{2x(k+m)+z(r(k+m)+2(r+m))}{r} \end{aligned} \tag{3.20}$$

In case a SN node is compromised in SHELL, k compromised keys are refreshed using m remaining keys that the compromised SN node does not know. So, the number of broadcast messages in the cluster by the CH node is m. However, the CH has to send k messages to the neighbouring CH nodes, which manage the k compromised keys. k messages are returned to the CH node with the new key values encrypted in the old ones. Then those k keys are aggregated and sent to the h neighbouring CH nodes, so that they can encrypt the aggregated message using all keys that they manage other than the k compromised keys. In turn, the CH receives m messages, which it broadcasts in its cluster. Therefore, average number of messages exchanged between CH nodes in case of SN node revocation in SHELL can be written as: -

$$Avg\_Msg\_Count\_Revoc\_SN^{SHELL}_{CH\rightarrow CH} = h+2k+m \tag{3.21}$$

Table 3.7: Average number of bytes transmitted by each type of node, in WSN using SHELL, LEAP+ and MUQAMI+, on each link in a cluster in case a sensor node is compromised in that cluster. All communications are within the cluster of the compromised sensor except $CH \rightarrow CH$ communication (* means a broadcast within cluster)

	MUQAMI+	**LEAP+**	**SHELL**
$CH \rightarrow CH$	-	-	$x(h+k+m)+zk$
$CN \rightarrow CH$	$3(x+z)((k+m)/r)$	-	-
$CH \rightarrow CN$	$3z((k+m)/r)$	-	-
$CH \rightarrow *$	-	x	xm
$CH \rightarrow SN$	$(2x(k+m)+z(r(k+m)+2(r+m)))/r$	xr	-
$SN \rightarrow CH$	$x(k/r)$	$x(d+1)(b/r)$	-
$SN \rightarrow SN$	-	$x(b+2d)(b/r)$	-
$SN \rightarrow *$	$x(m/r)$	x	-

where $Avg\_Msg\_Count\_Revoc\_SN^{SHELL}_{CH \rightarrow CH}$ is the average message exchanges between CH nodes when a SN node is compromised in SHELL. In Table 3.7, I compare communication overhead of my scheme with other schemes in case of SN node compromise. In LEAP+, if a SN node is compromised, only its neighbours perform the $SN \rightarrow CH$ and $SN \rightarrow SN$ communication. So, in order to calculate average message count, I multiply expressions for the count of both such messages by b/r.

Note that in all comparisons, there are no inter-cluster communications in my scheme as opposed to SHELL and there are no unicast communications among SN nodes as opposed to LEAP+. There is communication between CH nodes and the CN node but it is minimal and not very frequent. Due to these factors, my scheme has lesser communi-

cation and computation overhead than other schemes, which we will further establish in Section 3.4.

3.4 Simulation results

For simulation, my proposed network architecture is similar to the one shown in Figure 2.1. I have compared my scheme with two other schemes SHELL [49] and LEAP+[47], which are the most appropriate ones for WSNs proposed so far according to the best of my knowledge. In SHELL, CH nodes need to contact the neighbouring CH nodes so I have assumed a total of 5 clusters ($g = 5$) with 412 nodes in each cluster i.e. $r = 412$. I have assumed $k = m = 6$, so that there are ample key combinations left for addition of new nodes in the network. Also, I have assumed $b = 10$ and $d = 0.5$ on average. In case of SHELL, each CH node divides the EBS matrix into four equal parts and shares one part with each of the other CH nodes i.e. $h = 4$. The neighbouring CH nodes in turn manage keys for the part of EBS matrix shared with them. Simulation was programmed in "Tools Command Language (tcl8.0)", which is used to program ns-2 simulations.

G. Xing et. al.[105] states that the range of data transmission from sensor node is between -20dBm to 10dBm. I have assumed that the maximum distance between CN and a CH or between two CH nodes is about ten times that of a cluster size. Moreover, the maximum cluster size is around ten times the maximum distance between two neighbouring nodes. In order to record the power consumed during message exchanges, I have assumed three power levels for message transmission: one for communication outside the cluster, one for communication with a node inside the same cluster and one for communication with a neighbouring node. Also, I assume that the power levels is directly proportional to the distance of communication.

CH nodes transmits at 10dBm (10 mW) to communicate outside the cluster and 0dBm (1 mW) to communicate within its cluster. SN and KG nodes transmit at 0dBm to communicate with the CH or to broadcast within the cluster. Communication with the neighbouring node within the cluster is done at -10dBm (0.1 mW). [52] suggests that the application level bandwidth in WSN is around 19.2kbps and [105] suggests that its around 6kbps. I have assumed the application level bandwidth to be 19.2kbps in my simulations. I also simulated keeping it at 6kbps and found similar results.

Power level of a node during message reception and computation phase is assumed to be 0.1 mW. It is important to assume a power level for computation because I have also taken into account computation costs in my simulation. I have assumed that MICA2 motes are used and they have ATMEGA128L CPU as mentioned in [52]. Further, I have assumed that MD5 hashing scheme is used and IDEA cipher algorithm is used. For 16 bytes, MD5 takes 1.45msec, encryption using IDEA takes 0.68msec and decryption using the same algorithm takes 2.42msec on ATMEGA128L CPU according to [106].

I have assumed key size and key-chain seed size to be 16 bytes in my simulation. [107] states that an 8MHz processor can generate 50,000 random bytes in one minute. ATMEGA128L CPU also has a speed of 8MHz. So, I have calculated the time to generate a key or seed value as 19.2msec according to the calculations of [107]. Lastly, I have assumed the key-chain length l to be 32 in my simulations.

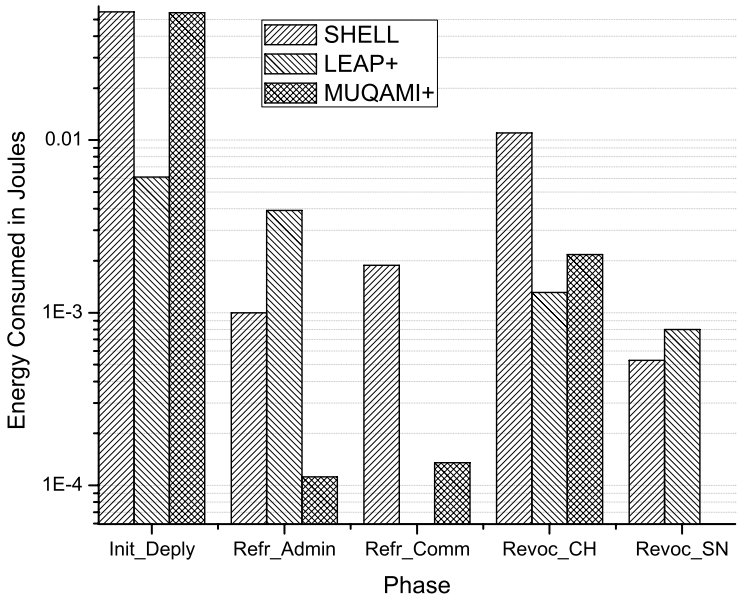
With the above set of simulation parameters, I recorded the average energy consumed by CH and SN nodes during these five phases: initial deployment, refreshment of administrative keys, refreshment of communication keys, revocation of a compromised CH node and revocation of a compromised SN node. Over 70 iterations were carried out for every phase. In case of MUQAMI+, I took the weighted average of SN and KG nodes and recorded it as average energy consumed by SN nodes. Weights were set according

to the number of number of KG nodes in a cluster i.e. $12/412$ in my case. I have plotted my graphs on logarithmic scale because of large differences in the readings recorded.

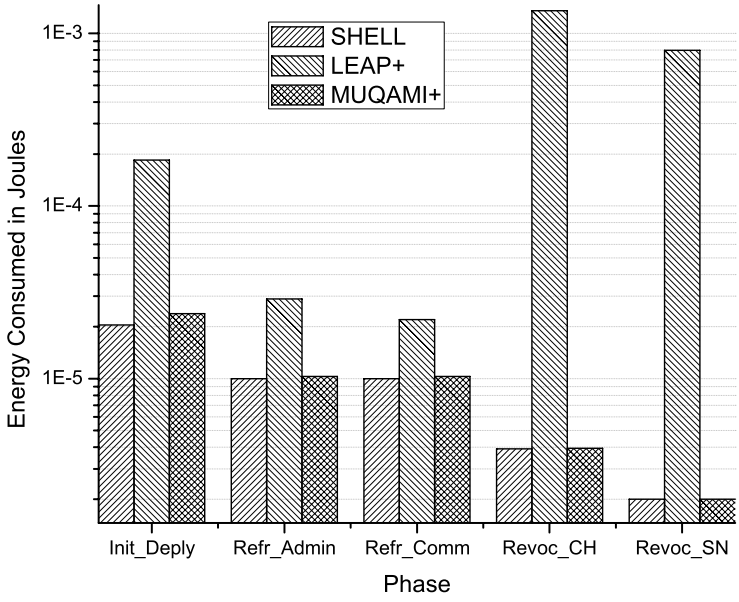
Figure 3.4(a) compares the average energy consumed by a CH node in each of the three schemes in all five phases. Apart from the initial deployment phase, in which SHELL and MUQAMI+ are comparable, MUQAMI+ outperforms SHELL in all other episodes by a comprehensive margin. MUQAMI+ outperforms LEAP+ in refreshment phase of admin key and revocation of a compromised SN node but the CH node in LEAP+ consumes less energy than the CH node in my scheme when considering the phases of initial deployment, refreshment of communication key and revocation of a compromised CH node. However, it comes at the cost of additional burden on the other SN nodes in those three phases as evident in Figure 3.4(b). Also, note that there is no additional burden on SN nodes of my scheme as compared to SHELL, whose CH nodes consume more energy as compared to my CH nodes.

In MUQAMI+, role of being a CH node can be transferred from one node to another from time to time. Also, the CH node of LEAP+ consumes less energy than that of mine and the SN node of my scheme consumes less energy than that of LEAP+. So I find it necessary to compare the average energy consumed by a node considering all type of nodes in the network. Figure 3.5 compares the average energy consumed by a node considering all types od nodes i.e. CH nodes and SN nodes together. Except for the initial deployment phase, my scheme outperforms both other schemes. MUQAMI+ outperforms LEAP+ due to the use of combinatorics and it outperforms SHELL due to the local distribution of key management responsibilities. Also, there is no single point of failure in my scheme.

Finally, I need to show the most significant improvement of my scheme and its scalability. For that purpose, I changed the number of nodes in my simulation code and then



(a) Cluster Head Node



(b) Sensor Node

Figure 3.4: Comparison of average energy consumed by different types of node in different phases of SHELL, LEAP+ and MUQAMI+ schemes in WSN

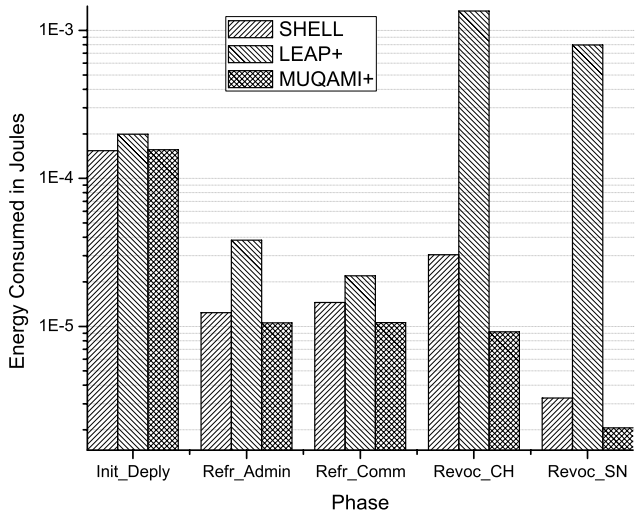
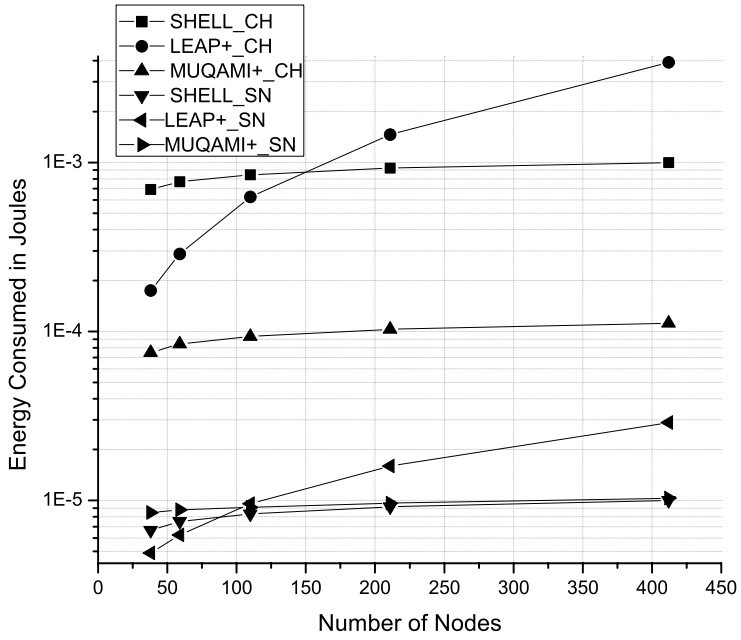
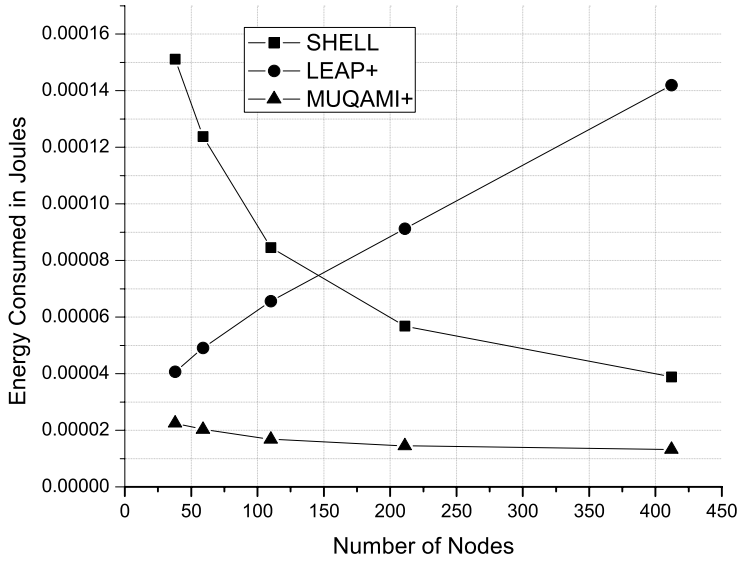


Figure 3.5: Comparison of Average Energy Consumed by a node in different phases of SHELL, LEAP+ and MUQAMI+ schemes in WSN

recorded the readings. Since the management of administrative key is most critical in key management, I have plotted, in Figure 3.6(a), trends of average energy consumed in the phase of administrative key refreshment by both type of nodes considering each of the three schemes while varying the number of nodes in the network. Figure 3.6(b) shows the average of both type of nodes. It is clear that my scheme is scalable as well as more efficient than the other two schemes. It is important to note the convergence of SHELL with MUQAMI+ in Figure 3.6(b). The improvement of my scheme over SHELL is in the energy consumed by the CH node. With increased network size, energy consumed by CH nodes averages with larger number of nodes. For LEAP+, I assume that the network density increases with the number of nodes. Similar trend is followed by all the schemes in communication key refreshment and node revocation phases.



(a) Plotting consumption of CH and SN nodes separately



(b) Plotting consumption of CH and SN nodes collectively (Average)

Figure 3.6: Comparison of Average Energy Consumed in Administrative Key Refreshment Phase of SHELL, LEAP+ and MUQAMI+ schemes in WSN with respect to the number of nodes in the network

3.5 Summary

In this work, I have proposed an EBS based key management scheme MUQAMI+, which also makes use of key-chains [101], for clustered sensor networks. Just like SHELL [49], MUQAMI+ is highly scalable, resilient against node capture and has effective node authentication mechanisms. Apart from that, it does not have single point of failure in a cluster or in a network. However, its mechanism of avoiding single point of failure is different and more efficient than that of SHELL scheme. Instead of relying on the neighbouring cluster head nodes for key management, responsibility of key management is distributed among few key generating nodes within a cluster. This reduces communication, computation, storage overhead and energy consumption of the sensor nodes in the network. A big advantage of this scheme is that it is very flexible i.e. it allows the responsibility of being cluster head node and being a key generating node to be shifted seamlessly from one node to another. Therefore, if this scheme is employed in a sensor network, responsibilities can be transferred among nodes according to their capabilities and energy levels.

Chapter 4

Key Management for WBAN

4.1 Introduction

Due to the fact that WBAN consist of sensor nodes, they have been considered similar to WSNs. Therefore, most of the related work is from WSN paradigm. The most simple key management solution for WBAN is key pre-distribution just like in WSN. However, lack of key refreshment is a greater problem in WBAN as compared to WSN. In WBAN, network lifetime may be indefinite because nodes' batteries can be replaced or recharged. Under such circumstances, periodic key refreshment becomes even more necessary.

Many schemes, which support key refreshment, have been proposed for WSN and can be applied to WBAN. Key management scheme of Riaz et. al. [83] requires the base station to provide public keys to the communicating nodes. Drawback of Riaz's scheme is that frequent communication with the cluster head node incurs significant communication overhead. Paek et. al.[102] base their scheme on regional and virtual groups. LEAP+ [47] is a localized scheme and one of the state-of-the-art solutions for WSN. Common drawback of Paek's scheme and LEAP+ is their assumption that the network is safe during some initial time period. Also, both these schemes are not designed for a scenario, in which all nodes are in communication range of each other.

[108] and [82] use asymmetric cryptography in WSN using Elliptic Curve Cryptog-

raphy. Apart from being designed for large scale sensor networks, both of these schemes move the additional burden of public key cryptography to the cluster head node. Such strategies should be avoided because the cluster head nodes also have limited battery and become single point of failure in case they are compromised. Another drawback of [82] is that it assumes network safety during some initial time period.

SHELL [49] and MUQAMI+ [51] (proposed in the previous section) are lightweight solutions and suit the resource constrained sensor nodes well. They also avoid single points of failure in sensor networks. Both these schemes are based on combinatorics and Exclusion Basis System (EBS) matrix [84]. MUQAMI+, which is an extended version of [109], improves performance by distributing the key management responsibilities locally. Also, it makes use of key-chains [101], which are based on Lamport's one-time passwords [103]. However, both these schemes are designed keeping in mind large scale nature of clustered WSN. When applied to small scale networks, their performances drop considerably. Also, EBS based key management schemes are prone to collusion attacks [110].

All of the above schemes are generally efficient in WSN scenarios but none of them makes use of the characteristics of WBAN applications. Also, their designs are overly complex for WBAN scenarios. Previously, researchers have focused on application characteristics of WBANs but their research has been limited to the usage of biometrics values as keys and authentication codes [35],[36]. Importance of the research of [35] and [36] is that they have substantially reduced the computation costs involved in generating keys. Also, some researchers have focused on eradicating the need for key exchange [37],[38],[39] assuming that two communicating nodes can sense same biometric at the same time and then apply error-correcting codes to agree on a secret key. Eradicating the need for key exchange eradicates communication costs involved in key man-

agement. Apart from time synchronization issues, these schemes add another constraint to the network: they require some sensor nodes to sense more than one biometric. Having multiple sensors in a sensor node increases the cost of sensor node and may not be practical in many WBAN scenarios. Authors in [48] have eradicated time synchronization issues by using photoplethysmogram (PPG) signals for key exchange. To study its efficiency, they have also implemented their scheme in hardware [88]. However, issue of multiple sensing still remains a challenge.

In this chapter, I propose a complete key management architecture BARI+, keeping in mind application characteristics and security requirements of WBAN. Also, the proposed scheme does not have time synchronization and multiple sensing issues. BARI+ is a distributed key management scheme, which makes use of key refreshment schedules to distribute key management responsibility among all nodes in a WBAN in a fair manner. All nodes in WBAN are able to take part in key management because nodes need not generate keys themselves. After presenting my scheme, its overhead is analyzed and compared with other state-of-the-art schemes. Apart from analyzing storage and communication overhead, security of my scheme is also analyzed.

Rest of this chapter is organized as follows: section 4.2 presents my scheme. Section 4.3 analyzes my scheme and compares it with other state-of-the-art key management schemes. Section 4.3 also analyzes security of my scheme. Section 4.4 presents simulation results. In the end, section 4.5 provides the summary of this chapter. In this paper I use many abbreviations and notations like WBAN for wireless body area networks. Refer to the list of abbreviations and notations presented in Table 4.1 for complete list of abbreviations and notations used in this chapter.

Table 4.1: List of Notations Used in Chapter 4

WSN	**Wireless Sensor Network**
$WBAN$	**Wireless Body Area Network**
MS	**Medical Server**
PS	**Personal Server**
SN^i	**Sensor Node i**
$K^i_{SN,MS}$	**Key shared between Node i and the MS. It is preloaded in every node and refreshed whenever it is used**
K^i_{bsc}	**Basic Key of Node i shared with the PS. It is preloaded in every node and is refreshed whenever it is used**
K_{comm}	**Communication Key**
K^i_{admin}	**Administrative Key i**
mi	**Message number i in a particular communication sequence**
$E_K\{A\|B\}$	**Values A and B are put together in a block/chunk and then the chunk is encrypted using Key K**

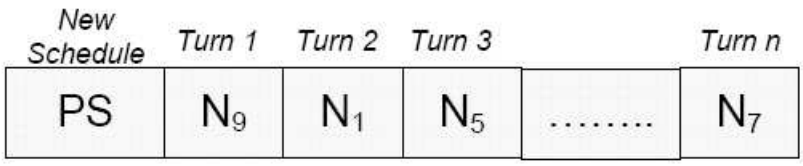


Figure 4.1: Example of key management schedule, of BARI+ scheme in WBAN, with n slots

4.2 BARI+

My scheme supports use of biometric measurements as symmetric keys because they posses randomness properties and can be used to generate symmetric keys in WBAN scenarios. This has already been discussed in previous sections. My scheme makes use of key refreshment schedule, which depicts the turn of each node for key refreshment. The personal server (PS) issues new key refreshment schedule periodically. Each node refreshes the key in the slot allotted to it. The PS can exempt some nodes from their key management responsibilities depending upon their energy level and transmission capabilities. Example of a key refreshment schedule is shown in Figure 4.1.

My scheme uses four types of keys to manage a WBAN: communication key, administrative key, basic key and a secret key shared between sensor node and the medical server. Communication key K_{comm} is a network wide key and is used to transfer data through the network in a secure manner. In my scheme, K_{comm} is managed by the PS itself. Since K_{comm} is used very frequently, it may come under cryptanalytic attacks and must be refreshed regularly.

Administrative key K_{admin} is used to refresh K_{comm}. K_{admin} is also a group key but it is not used as frequently as K_{comm}. Naturally, K_{admin} is less exposed as compared to K_{comm}. Although PS is more capable than a sensor node, PS is also a battery powered

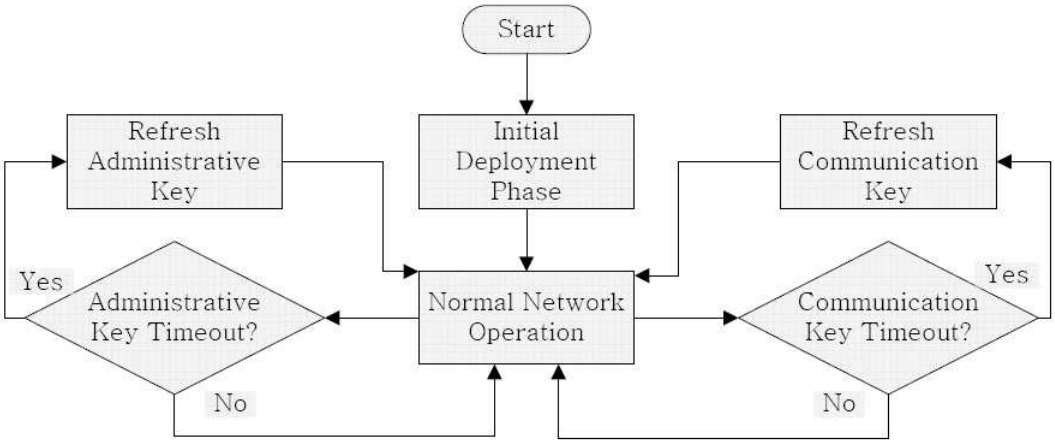


Figure 4.2: Flowchart of BARI+ scheme for WBAN excluding compromised node revocation

device. Also, sensor nodes need not generate keys in order to refresh them. Therefore, I use refreshment schedules to distribute the responsibility of key management evenly throughout the network. In order to increase resilience in a WBAN, one can increase the number of administrative keys being used. Figure 4.2 shows the manner, in which my scheme manages keys of WBAN when compromised node eviction is not done through software.

In WBAN applications, it is almost impossible for an adversary to compromise a node physically because of human presence. Although it is possible, it is less likely that an adversary can place a malicious node nearby to hack into a node's system software. Even if such an event occurs, it is a lot easier to detect as compared to WSN because the PS can directly monitor the activity of a compromised node and the compromised can be removed through human intervention. Despite the fact that there are lesser chances of malicious activities in WBAN, it is important to cover all possibilities. Also, K_{admin} needs to be refreshed through some other key at some point in time. Therefore, I employ

basic keys K_{bsc} in my key management framework. Every node has its own K_{bsc}, which it shares only with the PS. Key $K_{SN,MS}$ is a rarely used backup key shared between sensor node and the medical server. $K_{SN,MS}$ is important and is essential to recover from the compromise of PS or K_{bsc}.

4.2.1 Initial Deployment

PS is deployed in the beginning. Throughout network lifetime, PS is connected with the medical server through an external secure communication channel, which may be the internet. PS comes pre-loaded with K_{admin}, K_{comm} and basic keys of all nodes that are to be deployed in the network. Also, identities and authentication codes of all nodes are pre-loaded in the PS. These codes are used to authenticate the sensor nodes. After the PS is deployed, sensor devices are deployed on various parts of the body. Sensor nodes come pre-loaded with authentication codes of all nodes in the network, K_{admin} and their respective K_{bsc} and $K_{SN,MS}$. Soon after deployment, every node sends discovery message to the PS as follows: -

$$m1 : \forall i \quad if \exists SN^i : SN^i \rightarrow PS : E_{K_{admin}}\{ID^i | Auth\_Code^i\}$$

In WBAN, some sensor nodes may have very little communication range. MS informs the PS about deployment of such nodes in advance. If such nodes are to be deployed, the PS commands other nodes to forward discovery messages of such nodes to the PS. After all the sensor nodes are deployed, the PS generates a key refreshment schedule for K_{admin}. It then broadcasts the refreshment schedule and initial value of K_{comm} in the network as follows: -

$$m2 : PS \rightarrow * : E_{K_{admin}}\{K_{comm} | Key\_Ref\_Schedule | Auth\_Code^{PS} | Timestamp\}$$

In order to prevent the PS from waiting forever, there is a timer. As soon as the last expected node's discovery message is received or the timer expires, the PS calculates the refreshment schedule and broadcasts its initial message $m2$. All subsequent nodes are treated as added nodes and deployed through the procedure explained in subsection 4.2.3.

4.2.2 Re-keying

In order to refresh K_{comm}, PS computes a value from biometrics as the value of new K_{comm}. It then encrypts the new value of K_{comm} with K_{admin} and broadcasts it into the network as follows: -

$$m1 : PS \rightarrow * : E_{K_{admin}}\{K_{comm}|Auth\_Code^{PS}\}$$

Administrative key is refreshed periodically. When the turn of sensor node i arrives, sensor node i waits for a certain period of time, computes a new value for K_{admin} from biometrics and broadcasts it in the network as follows: -

$$m1 : SN^i \rightarrow * : E_{K^{old}_{admin}}\{K^{new}_{admin}|Auth\_Code^i\}$$

When the key refreshment schedule expires, the PS calculates the new schedule, encrypts it in the current value of K_{admin} and broadcasts it into the network as follows: -

$$m1 : PS \rightarrow * : E_{K_{admin}}\{Key\_Ref\_Schedule|Auth\_Code^{PS}|Timestamp\}$$

When a network is deployed, key refreshment timeout of every sensor node is decided according to pre-defined criteria. However, PS can decide to refresh K_{admin} at any point in time if it detects malicious activities. In such scenario, the PS sends key refresh message to the node, which is supposed to refresh K_{admin} next time. For example, if it

is the turn of sensor node i to refresh the administrative key, following messages will be exchanged to refresh K_{admin}: -

$$m1 : PS \rightarrow SN^i : E_{K_{admin}}\{Key\_Refresh\_Msg|Auth\_Code^{PS}|Timestamp\}$$

$$m2 : SN^i \rightarrow * : E_{K^{old}_{admin}}\{K^{new}_{admin}|Auth\_Code^i\}$$

In order to maintain forward secrecy, K_{admin} needs to be refreshed through K_{bsc} regularly. Also, K_{admin} needs to be refreshed through K_{bsc} in case of sensor node compromise. In order to refresh K_{admin} through K_{bsc}, the PS computes new values of basic keys and refreshes K_{admin} using K_{bsc} of the sensor nodes: -

$$m1 : \forall i \quad if \exists SN^i : PS \rightarrow SN^i : E_{K^i_{bsc\_old}}\{K_{admin}|K^i_{bsc\_new}$$
$$|Auth\_Code^i_{new}|Auth\_Code^{PS}\}$$
$$m2 : PS \rightarrow * : E_{K_{admin}}\{K_{comm}|Auth\_Code^{PS}\}$$

Although basic keys are used only once and refreshed after every use, it is possible that they need to be refreshed using some other key. For example, if the PS is compromised. Therefore, I think it is important to have a procedure to recover from such catastrophic failures. In such scenario, authentication code of the PS is also refreshed. If a new PS is deployed, it comes pre-loaded with K_{admin} and K_{comm}. MS sends identities, authentication codes and basic keys of the sensor nodes to the newly deployed PS. If the PS is not replaced, MS sends new values of K_{bsc} to the PS. Also, MS encrypts new values of K_{bsc}, along with the new authentication code of PS, in $K_{SN,MS}$ of all sensor nodes and sends them to the PS through an external secure communication channel, which may be the internet. After receiving messages encrypted in $K_{SN,MS}$ of the sensor nodes, PS just forwards them to the respective sensor nodes. Whenever K_{bsc} is refreshed, K_{admin} and K_{comm} are also refreshed. Following message exchanges take place between the PS and

sensor nodes when K_{bsc} is refreshed using $K_{SN,MS}$: -

$$m1 : \forall i \quad if \exists SN^i : PS \rightarrow SN^i : E_{K^i_{SN,MS\_old}}\{K^i_{bsc} | Auth\_Code^{PS}_{new}$$
$$| K^i_{SN,MS\_new} | Auth\_Code^{MS}\}$$
$$m2 : \forall i \quad if \exists SN^i : PS \rightarrow SN^i : E_{K^i_{bsc\_old}}\{K_{admin} | K^i_{bsc\_new}$$
$$| Auth\_Code^i_{new} | Auth\_Code^{PS}\}$$
$$m3 : PS \rightarrow * : E_{K_{admin}}\{K_{comm} | Auth\_Code^{PS}\}$$

Note that $K_{SN,MS}$ is refreshed whenever it is used. Also, the PS does not get to know key $K_{SN,MS}$ of any sensor node. Remaining key refreshment schedule is followed after the refreshment of K_{admin} irrespective of the way K_{admin} is refreshed.

4.2.3 Node Addition

In some cases, new nodes are added to the network or the existing nodes are replaced. One possible scenario of node addition can be the deployment of a new device to monitor some biometric. Similarly, one possible scenario of node replacement is malfunction of a device. Under such circumstances new nodes are added to the network.

If new nodes are to be added in the network, MS informs PS about new deployments by sending identities, basic keys and authentication codes of new nodes to the PS. Also, MS informs the PS about initial value of K_{admin} that is preloaded into the new nodes. All this communication is done through an external secure communication channel. Under normal circumstances, if the PS receives messages from stranger nodes, it ignores them and indicates malicious activity on its own output. If informed by the MS, the PS expects discovery messages from new nodes. This is important because otherwise malicious nodes can drain its energy by sending fake discovery messages. New nodes send their

respective discovery messages encrypted in the pre-loaded value of K_{admin} as follows: -

$$m1 : \forall SN^j \in \{New\_Nodes\} : SN^j \rightarrow PS : E_{K_{admin}^{pre-load}}\{ID^j | Auth\_Code^j\}$$

If nodes, which have very limited communication range, are deployed, then the PS commands other sensor nodes to forward their discovery messages to the PS. PS waits for all expected nodes for a certain period of time. After that, it broadcasts the remaining key refreshment schedule and current values of K_{comm} and K_{admin} to newly deployed nodes as follows: -

$$m2 : PS \rightarrow * : E_{K_{admin}^{pre-load}}\{K_{comm} | K_{admin} | Remaining\_Sched$$
$$| Auth\_Code^{PS} | Timestamp\}$$

All nodes, except the newly deployed ones, ignore such message from the PS. Newly deployed nodes can participate in key refreshment procedure after the next key refreshment schedule is issued by the PS.

4.3 Analysis and Comparison

In this section, I establish my claims regarding efficiency of my scheme BARI+ by analyzing its storage and communication overheads and comparing it with other key management schemes. Also, security analysis of my scheme is presented at the end of this section. According to my knowledge, this is the first key management scheme that is proposed for WBAN and does not require multiple sensing. Therefore, I compare my scheme with two other state-of-the-art key management schemes, which are designed for WSN, LEAP+ [47] and MUQAMI+ [51]. SHELL [49] is also a state-of-the-art key management scheme for WSN but it is not applicable to WBAN because it requires services of neighbouring cluster head nodes, which may not be present in WBAN scenario.

When applying LEAP+ and MUQAMI+ in WBAN scenario, I assume that the PS acts as cluster head node and all nodes on one body are part of the same cluster. Also, cluster can not span multiple bodies.

4.3.1 Storage Overhead

Storage and exchange of authentication codes is common in all key management schemes. Also, storage requirements of authentication codes do not make much difference when key management schemes are compared with respect to their storage overhead. For simplicity, storage requirements of authentication codes is not included in storage analysis. Considering storage overhead of sensor nodes, only four keys are stored: K_{comm}, K_{admin}, K_{bsc} and $K_{SN,MS}$. Apart from that, key refreshment schedule is stored on sensor nodes. A sensor node keeps track of its turn with the help of two short integers. One integer contains a counter to keep track of its turn. The other one indicates timeout after which it refreshes K_{admin}. If a short integer requires 2 bytes and key length is z bytes, Then the storage requirement of a sensor nodes becomes: -

$$SR_{SN}^{BARI+} = (4 \times z) + 4 \tag{4.1}$$

PS stores K_{bsc} of all sensor nodes, K_{admin} and K_{comm}. Also, it stores complete key refreshment schedule for K_{admin}. Storing a sensor node's identity requires 2 bytes. Another 2 bytes are required to specify timeout after which sensor node refreshes K_{admin}. So, the storage requirements of PS becomes: -

$$SR_{PS}^{BARI+} = ((r + 2) \times z) + (4 \times r) \tag{4.2}$$

where r is the number of nodes in WBAN formed on a body. Note that key $K_{SN,MS}$ is not stored on the PS.

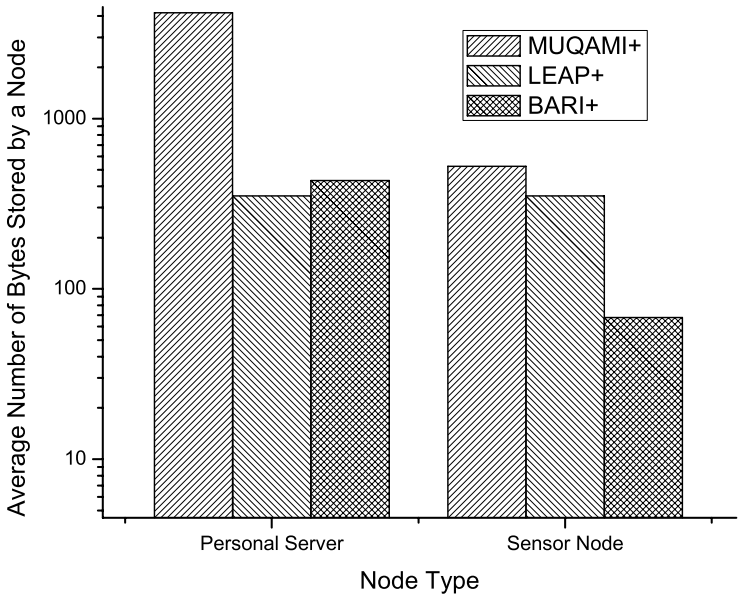


Figure 4.3: Comparison of Storage Requirements of MUQAMI+, LEAP+ and BARI+ schemes in WBAN

Storage requirements of a node (sensor node or personal server) in LEAP+ is fairly straightforward. Apart from pairwise keys shared with each node in its cluster, every node stores its cluster key and the communication key. So, the storage requirement of a node in LEAP+ becomes: -

$$SR_{PS \vee SN}^{LEAP+} = z \times (r + 2) \tag{4.3}$$

In MUQAMI+, each PS node has to store K_{comm} and $K_{cn,ch}$. Also, PS has to store $K_{ch,sn}$ of all SN nodes and key-chains of key $K_{ch,kg}$ of all KG nodes in its cluster. In addition to that, PS has to store EBS matrix. If EBS data for each node takes 4 bytes (2 bytes for storing node identity and 2 bytes for storing key pattern), it takes $4 \times r$ bytes to store EBS matrix. So, the average storage requirement of a PS node (in bytes) of

Table 4.2: Storage requirements (in bytes) of each type of node using MUQAMI+, LEAP+ and BARI+ schemes in WBAN

	Personal Server	**Sensor Node**
MUQAMI+	$(z \times ((l \times (k+m)) + r - (k+m) + 2)) + (4 \times r)$	$(z \times ((k+4) + [(2 \times (l-1) \times (k+m))/r]))$
LEAP+	$z \times (r+2)$	$z \times (r+2)$
BARI+	$((r+2) \times z) + (4 \times r)$	$(4 \times z) + 4$

MUQAMI+ becomes: -

$$SR_{PS}^{MUQAMI+} = (z \times ((l \times (k+m)) + r - (k+m) + 2)) + (4 \times r) \tag{4.4}$$

where l is the length of key-chains [101], which are used by MUQAMI+ for key management and k and m are EBS [84] parameters. In MUQAMI+, SN nodes have to store k admin keys apart from four other keys: $K_{ch,sn}$, K_{comm}, K_{bsc} and K_{disc}. So, the average storage requirement of a sensor node in MUQAMI+ can be expressed as: -

$$SR_{SN}^{MUQAMI+} = z \times (k+4) \tag{4.5}$$

Among sensor nodes, MUQAMI+ also has key generating (KG) nodes, which store two key-chains: one for the admin key, which it generates and one for $K_{ch,kg}$. Also, KG nodes store $k-1$ EBS keys along with three other keys: K_{comm},K_{bsc} and K_{disc}. So, the average storage requirement of a KG node can be expressed as: -

$$\begin{aligned} SR_{KG}^{MUQAMI+} &= z \times (2 \times l + (k-1) + 3) \\ &= z \times (2 \times (l+1) + k) \end{aligned} \tag{4.6}$$

Table 4.3: Average number of messages transmitted by each type of node in initial deployment phase of MUQAMI+, LEAP+ and BARI+ schemes in WBAN

	Personal Server	**Sensor Node**
MUQAMI+	r	1
LEAP+	$2 \times (r+1)$	$2 \times r + 1$
BARI+	1	1

In MUQAMI+, there are $k+m$ KG nodes out of a total of r nodes in a cluster. Therefore, average storage requirement of each node within a cluster comes out to be: -

$$\begin{aligned} SR_{SN \cup KG}^{MUQAMI+} &= z \times \frac{(r-(k+m))(k+4)+(k+m)(2(l+1)+k)}{r} \\ &= z \times \frac{r \times (k+4)+(k+m) \times (2 \times (l+1)-4)}{r} \\ &= z \times ((k+4)+\frac{2 \times (l-1) \times (k+m)}{r}) \end{aligned} \tag{4.7}$$

Note that $(k+m) << r$ only for large scale networks. For small scale networks like WBAN, k and m are comparable to r, which degrades the performance of MUQAMI+ considerably.

Table 4.2 compares the storage requirements of BARI+ with MUQAMI+ and LEAP+. It is clear from table 4.2 that overall storage overhead of our scheme is less as compared to other schemes. To strengthen our claim, we have also compared storage requirements of all three schemes in Figure 4.3. In Figure 4.3, we have assumed $k = m = 4$, $l = 32$, $r = 20$ and $y = 1$.

Table 4.4: Average number of messages transmitted by each type of node using MUQAMI+, LEAP+ and BARI+ schemes when communication key is refreshed in WBAN

	Personal Server	**Sensor Node**
MUQAMI+	$k+m$	$(k+m)/r$
LEAP+	1	–
BARI+	1	–

4.3.2 Communication and Computation Overhead

Communication is the most energy consuming activity in WBAN. Since all nodes are in communication range of each other, one only needs to analyze average number of messages transmitted by each type of node in every phase. Initial deployment phase of BARI+ is very lightweight and simple because every node has to send one message each.

Initial deployment phase of MUQAMI+ is also simple. Every sensor node has to send one discovery message each. In return, the PS has to send one message to each node in the network, which makes the total number of messages transmitted by the PS equal to r. In LEAP+'s initial deployment phase, the PS has to send one broadcast message to all nodes in the network. All nodes reply and pair-wise keys are established. After that, it sends its cluster key to each of the r nodes one by one and then broadcasts its group key in the network. Also, it replies to the initial messages sent by other nodes. So, the average number of messages transmitted by PS in the initial deployment phase

Table 4.5: Average number of messages transmitted by each type of node using MUQAMI+, LEAP+ and BARI+ schemes when administrative key is refreshed in WBAN

	Personal Server	**Sensor Node**
MUQAMI+	$(k+m) \times (1+(1/l))$	$((k+m)/r) \times (1+(1/l))$
LEAP+	r	r
BARI+	$y+((y+1)/r)$	$1/r$

of LEAP+ becomes: -

$$\begin{aligned} Avg\_Msg\_Count\_Init_{PS}^{LEAP+} &= (2 \times r) + 2 \\ &= 2 \times (r+1) \end{aligned} \tag{4.8}$$

Sensor nodes do not have to broadcast the communication key. Therefore, average number of messages transmitted by sensor nodes in the initial deployment phase of LEAP+ becomes: -

$$Avg\_Msg\_Count\_Init_{SN}^{LEAP+} = (2 \times r) + 1 \tag{4.9}$$

Comparison of my scheme with MUQAMI+ and LEAP+ is given in table 4.3, which indicates that my scheme BARI+ is more efficient as compared to other schemes. Nodes are added in the same way as they are initially deployed.

In BARI+, PS broadcasts one message to refresh communication key. Similarly, PS broadcasts one message in the network to refresh communication key in LEAP+ too. Both in BARI+ and LEAP+, sensor nodes need not send any message to refresh communication key. In MUQAMI+, PS sends $k+m$ messages to the key generating nodes, which in turn broadcast one message each. So, average number of messages

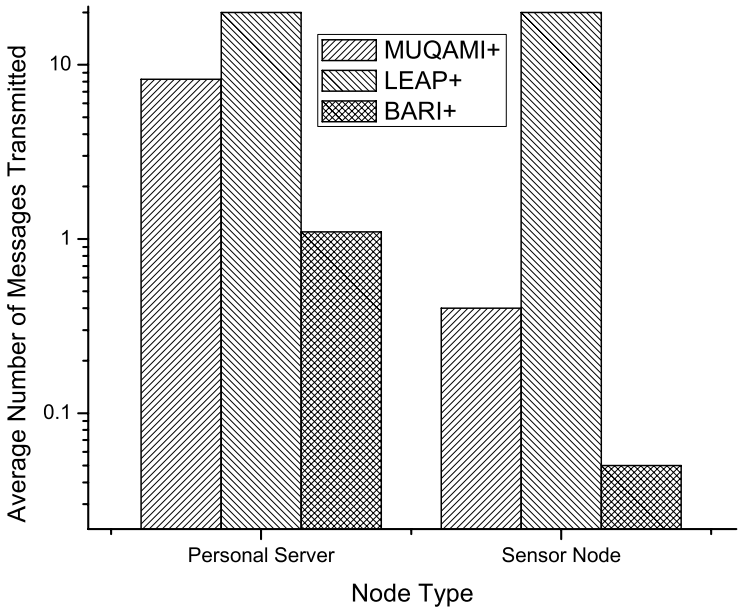


Figure 4.4: Comparison of Average Number of Messages Transmitted to Refresh Admin Key in MUQAMI+, LEAP+ and BARI+ schemes in WBAN

transmitted by a sensor node for communication key refreshment is expressed as: -

$$Avg\_Msg\_Count\_Rekey\_Comm_{SN}^{MUQAMI+} = \frac{k+m}{r} \tag{4.10}$$

Comparison of communication overhead for refreshment of communication key is given in table 4.4.

$$Avg\_Msg\_Count\_Rekey\_Admin_{PS}^{MUQAMI+} = (k+m) \times (1 + (\frac{1}{l})) \tag{4.11}$$

Refreshment of administrative key is also lightweight in my scheme. In order to refresh administrative key, each node sends one message in every schedule. If all nodes participate in administrative key refreshment, average number of messages sent by each node to refresh K_{admin} one time comes out to be $1/r$. However, K_{admin} is also refreshed

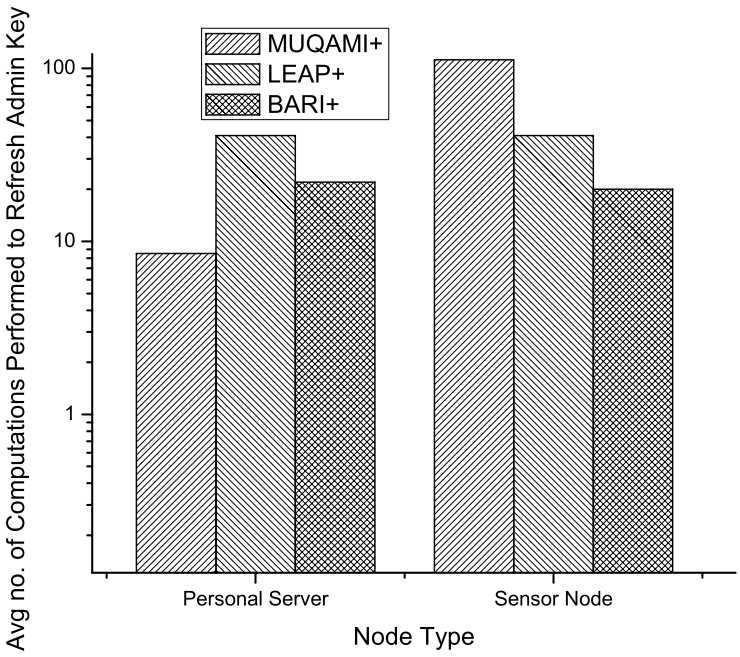


Figure 4.5: Comparison of Average Number of Computations Performed to Refresh Admin Key in MUQAMI+, LEAP+ and BARI+ schemes in WBAN

by K_{bsc}. If refreshed through K_{bsc}, PS sends two messages for the purpose. If K_{admin} is refreshed by K_{bsc} y times in every key refreshment schedule, then average number of messages transmitted by PS for administrative key refreshment becomes: -

$$Avg\_Msg\_Count\_Rekey\_Admin_{PS}^{BARI+} = y + \frac{y+1}{r} \tag{4.12}$$

In LEAP+, every node has to send one message to each of r other nodes in the network. In MUQAMI+, PS has to send $k+m$ messages to key generating nodes and one message after every l key refreshments to get new seed values for key-chains. However, KG nodes can use biometric values in case of WBAN. So, average number of messages transmitted by PS for refreshment of administrative key in MUQAMI+ becomes: -

Similarly, average number of messages transmitted by a sensor node for refreshment

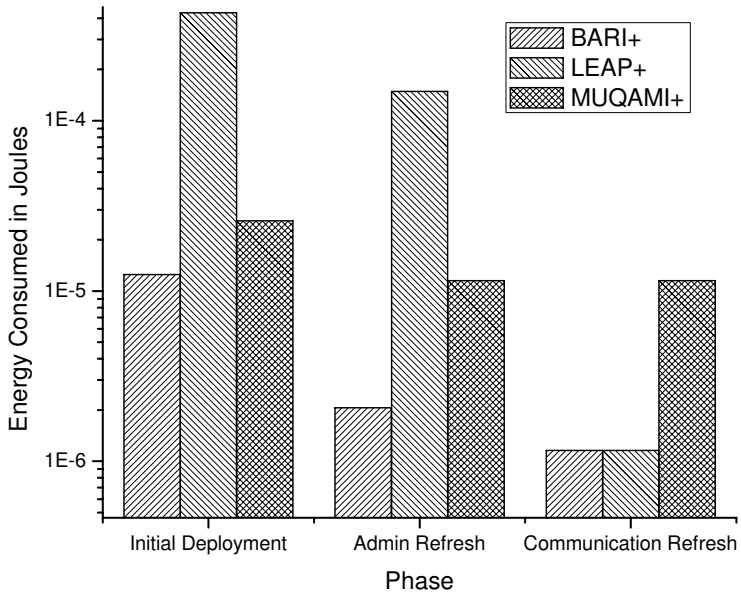


Figure 4.6: Comparison of Average Energy Consumed by a Sensor Node in different phases of MUQAMI+, LEAP+ and BARI+ schemes in WBAN

of administrative key in MUQAMI+ comes out to be: -

$$Avg\_Msg\_Count\_Rekey\_Admin_{SN}^{MUQAMI+} = (\frac{k+m}{r}) \times (1 + (\frac{1}{l})) \qquad (4.13)$$

Table 4.5 compares my scheme BARI+ with MUQAMI+ and LEAP+ in administrative key refreshment phase. Also, Figure 4.4 and Figure 4.5 show the average number of messages transmitted by each type of node and average number of computations performed by each type of node respectively. These figures show that overall performance of BARI+ is better than LEAP+ and MUQAMI+. In Figure 4.4 and Figure 4.5, we have assumed $k = m = 4$, $l = 32$, $r = 20$ and $y = 1$.

4.4 Simulation Results

In my simulations, assumed network architecture was similar to the one shown in Figure 2.2. My scheme BARI+ is compared with two other schemes MUQAMI+ [51] and LEAP+[47], which are state-of-the-art key management schemes for WSNs. MUQAMI+ uses EBS matrices and key-chains. EBS parameters k and m were assumed to be $k = m = 4$, so that ample key combinations are left for addition and replacement of nodes in the network. Also, key-chain length in MUQAMI+ was assumed to be 32 so that both storage and communication costs can be kept within practical limits. Number of sensor nodes was assumed to be 15 and key size was assumed to be 16 bytes in my simulations. Moreover, it was assumed that in BARI+, K_{admin} is refreshed through K_{bsc} every time a key refreshment schedule expires. Simulations were programmed in "Tools Command Language (tcl8.0)", which is used to program ns-2 simulations.

My scheme uses biometrics as keys and need not generate them but other schemes were not designed to take advantage of this property of WBANs. Therefore, cost of key generation is also included in my simulations. [107] states that an 8 MHz processor like ATMEGA128L CPU can generate 50,000 random bytes per minute. According to [107], generating a key or a seed value takes 19.2 msec on 8 MHz processor.

According to G. Xing et. al. [105], range of data transmission of a sensor node is between -20dBm and 10dBm. In WBAN scenario, all nodes are nearby and the ones, participating in key management, are in communication range of each other. Therefore, only one power level was assumed for message transmission. In my simulations, transmission power level was assumed to be 0 dBm (1 mW). Power level during reception and computation phases was assumed to be -10 dBm (0.1 mW). Power level for computation phase was included because computation costs were included considered in my simulations. Usage of MICA2 motes, which have ATMEGA128L CPU as mentioned in [52],

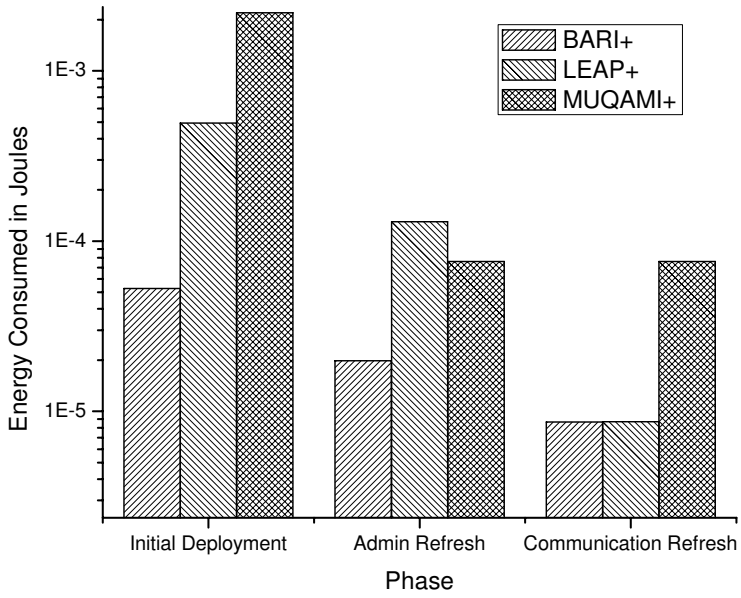


Figure 4.7: Comparison of Average Energy Consumed by a Personal Server in different phases of MUQAMI+, LEAP+ and BARI+ schemes in WBAN

was assumed. Moreover, usage of SHA1 hashing scheme and RC5 cipher algorithm was assumed. According to [106], hashing for 16 bytes using SHA1 algorithm takes approximately 3.7 msec; both encryption and decryption for the same length of data using RC5 algorithm takes approximately 3.25 msec on ATMEGA128L CPU.

Apart from power levels, bandwidth of transmission link needs to be consideration. [52] suggests that the application level bandwidth in WSN is around 19.2 kbps whereas [105] suggests that it is around 6 kbps. In my simulations, application level bandwidth was assumed to be 19.2 kbps. Similar results were found when simulations, assuming application level bandwidth to be 6 kbps, were performed.

With the above set of simulation parameters, average energy consumed by PS and SN nodes during initial deployment phase, administrative key refreshment phase and communication key refreshment phase was recorded. For each phase, my simulations

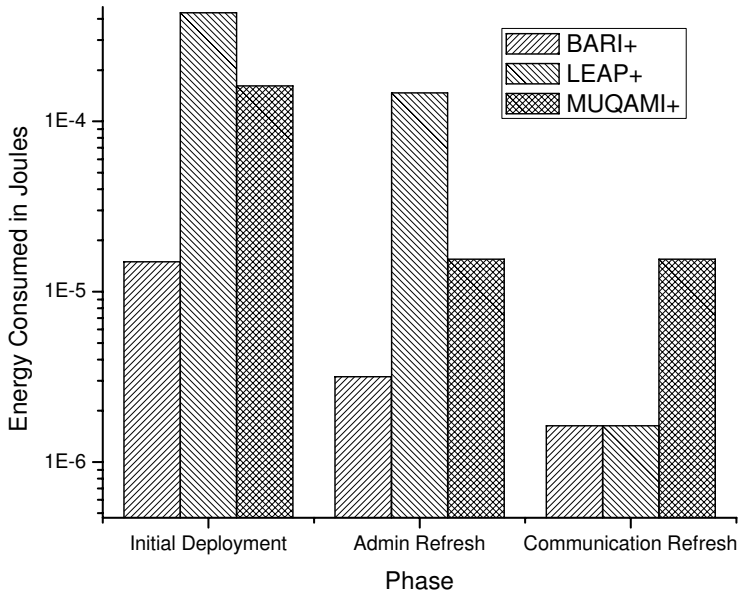


Figure 4.8: Comparison of Average Energy Consumed by a Node (including Sensor Nodes and the Personal Server) in different phases of MUQAMI+, LEAP+ and BARI+ schemes in WBAN

had more than 70 iterations. For MUQAMI+, weighted average of sensor nodes and key generating nodes was recorded as average energy consumed by SN nodes. Weights were set according to the number of number of key generating nodes in a network i.e. $k+m = 8$ in my case. Graphs are plotted on logarithmic scale because of large differences in readings of different schemes.

Figure 4.6 compares the average energy consumed by a sensor node in each of the three schemes in all three phases. My scheme proves to be more efficient than MUQAMI+ in all the three phases and better than LEAP+ in initial deployment and administrative key refreshment phase. I observe similar results when I compare the average energy consumption of a personal server in each of the three schemes in all three phases in figure 4.7. Figure 4.8 compares the average energy consumed by a node (tak-

ing into account sensor nodes and the personal server) in each of the three schemes in all three phases.

4.5 Summary

In this work, I have proposed a scheme that covers shortcomings of existing key management solutions for WBAN. Apart from time synchronization and other issues in error correcting codes [39], sensor nodes are supposed to sense multiple biometrics in existing key management schemes for WBAN [48]. This is not always possible because a patient any other human being might refuse to wear more than a certain number of devices. Also, one device is used to measure one biometric most of the time. Devices, measuring multiple biometrics have financial implications. Proposed scheme do not require sensor nodes to sense multiple biometrics. Also, it does not have issues like time synchronization. Apart from that, proposed scheme outperforms other schemes, designed for WSN because it is designed keeping in mind scale, topology and application characteristics of WBAN.

Chapter 5

Security Analysis

Analyzing security of a key management scheme is of utmost importance. A key management scheme may not prove to be effective against all attacks that can take place in a network. However, it is important to establish that a key management scheme is effective against the attacks, which can be handled by a key management scheme. At least, a key management scheme should be effective against the attacks, for which it is designed. A key management scheme is not very useful if it does not fulfill security requirements, which be be fulfilled by a key management scheme, of the target network. In this chapter, we discuss how each of our schemes, MUQAMI+ and BARI+, is effective against the possible attacks in WSN and WBAN respectively. Also, we compare with other state-of-the-art schemes in terms of security as we did in Chapter 3 and Chapter 4.

5.1 MUQAMI+ (Wireless Sensor Networks)

In MUQAMI+ scheme for wireless sensor networks, we have four types of keys: communication key, administrative keys, pair-wise keys between sensor nodes and the command node and pair-wise keys between sensor nodes and cluster head nodes. I agree with the group of researchers ([49],[47]), who think that a combination of different types of keys should be used in WSN because each type of key has a significance in maintaining

security and/or ensuring that the target network works normally.

From security point of view, it is ideal that all communicating nodes in WSN have pair-wise keys amongst them. However, it is disastrous for normal working of WSN because of excessive storage, computation and communication overhead. Usage of pair-wise keys hampers in-network processing, data aggregation and energy conservation due to message overhearing.

Using a single group key in a cluster makes most use of in-network processing, data aggregation and energy conservation due to message overhearing because all of the concerned nodes can decipher messages being sent towards the command node. However, repetitive use of a single group key for communication makes the key more vulnerable to cryptanalytic attacks and compromise of the key or a single node compromises the whole cluster. In order to avoid cryptanalytic attacks, the single group key must be refreshed regularly. Also, it should be refreshed using some other key to maintain forward secrecy.

If single group key is refreshed using pair-wise keys, it is not scalable and energy efficient because the cluster head node will have to send separate message to each node in its cluster for key refreshment. In order to achieve scalability and energy efficiency, I use EBS-based group keys to refresh the single group key. In my scheme, use of EBS-based group keys ensure scalability and energy efficiency along with forward secrecy. Also, use of EBS-based group keys ensure that compromise of a single key or node does not compromise the whole cluster. It is the use of EBS-based group keys for key refreshment and node revocation and management of EBS-based group keys locally i.e. within a cluster that makes MUQAMI+ scalable and more efficient than other schemes.

Despite the fact that single group key is efficient to use in WSN and EBS-based group keys can be used to refresh single group key, it is important that sensor nodes

share pair-wise keys with their respective cluster head nodes so that they can report malicious activities and conceal their reports from malicious nodes. Also, sensor nodes must share pair-wise keys with the command node to ensure secure initial deployment and secure replacement of a compromised cluster head node. Therefore, we use pair-wise keys between sensor nodes and their cluster head nodes and pair-wise keys between sensor nodes and the command node along with EBS-based group keys, which we call administrative keys and a single group key in a cluster, which we call communication key. In this section, I present security analysis of MUQAMI+ scheme with respect to vulnerabilities and attack vectors discussed in chapter 2. Under each vulnerability, each attack vector and effectiveness of MUQAMI+ under that vector is discussed. Also, we compare MUQAMI+ with other key management schemes in terms of security.

5.1.1 Passive Listening

MUQAMI+ is a key management scheme that involves encryption/decryption using secret keys. Therefore, adversaries can not listen to private communication. Also, MUQAMI+ is not susceptible to traffic pattern analysis as it allows responsibilities of being key generating node and being cluster head node to be regularly shifted from one node to another. Apart from that, MUQAMI+ has a reliable re-keying mechanism to avoid cryptanalytic attacks. Following are the attacks that can take place due to passive listening and details of how MUQAMI+ defends against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.1.1 in chapter 2) Adversary can listen to the private communication between two sensor nodes so that confidentiality of information is breached.

Counter Measure: In order to maintain confidentiality of private information,

MUQAMI+ scheme uses secret keys for all communications. Initial deployment of nodes is carried out using pair-wise keys between sensor nodes and the command node. Then administrative keys, pair-wise keys between sensor nodes and cluster head nodes and communication keys are established. All application related information is encrypted using communication keys. Messages that report malicious activity are encrypted using pair-wise keys between sensor nodes and cluster head node and messages related to the refreshment of communication keys are encrypted in administrative keys or pair-wise keys. For example, a SN node communicates application data to its CH node in the following manner: -

$$message : SN \rightarrow CH : E_{K_{comm}}\{Application\_Data\}$$

Refer to Section 3.2 in chapter 3 to see how each message exchanged in MUQAMI+ scheme is encrypted using secret keys.

Attack 2: (See enumeration 2 in section 2.3.1.1 in chapter 2) Even if messages are encrypted, adversary can analyze traffic patterns, which may lead to compromise of an important node.

Counter Measure: In MUQAMI+, compromise of any single node does not compromise the whole cluster i.e. none of the nodes is too important. Also, MUQAMI+ allows shuffling of responsibilities among sensor nodes regularly in an efficient manner so that none of the nodes seem more important than other nodes to an adversary. An administrative key is known to a group of sensor node, one of which is a key generating (KG) node and has the responsibility of refreshing it. To award responsibility to some other node in the group, CH node will need to send just a seed value to shift responsibility as follows: -

$$message : CH \rightarrow New\_KG : E_{K_{ch,sn}}\{New\_Seed\_Value\}$$

Table 5.1: Comparison of Defense Against Attacks due to Passive Listening in WSN

	SHELL	**LEAP+**	**MUQAMI+**
Breach in Confidentiality	Yes	Yes	Yes
Cryptanalysis	Yes	Yes	Yes
Single Point of Failure	Yes	Yes	Yes
Traffic Analysis	No	No	Yes

To shift responsibility of being cluster head node, only the EBS matrix needs to be shared between different nodes.

Attack 3: (See enumeration 3 in section 2.3.1.1 in chapter 2) Adversary can carry out cryptanalytic attacks to reveal secret keys.

Counter Measure: To prevent from cryptanalytic attacks, all secret keys are refreshed regularly. Communication keys are refreshed using administrative keys and, if required, pair-wise keys can be used to refresh administrative keys. For example, following message exchanges take place when KG node j, in cluster i refreshes administrative key l: -

$$m1 : CH^i \rightarrow KG^j : E_{K^{i,j}_{ch,kg}}\{Refresh\_Message\}$$

$$m2 : KG^j \rightarrow * : E_{K^l_{admin}}\{K^l_{admin\_new}\}$$

For complete detail on key refreshment procedures, refer to section 3.2.2 in chapter 3.

If we consider security of other state-of-the-art key management schemes for WSN SHELL [49] and LEAP+ [47] against passive listening, we find that both of these schemes provide defense against breach in confidentiality and cryptanalytic attacks as

both key management schemes employ encryption using secret keys and both provide for key refreshment at regular intervals. Apart from that, they also do not have single points of failure in the network. However, SHELL and LEAP+ do not allow responsibilities to be shifted from one node to another during network operation. Therefore, an adversary can easily figure out important nodes in the network by analyzing the traffic patterns. Comparison of our scheme MUQAMI+ with the other two schemes is shown in Table 5.1.

5.1.2 Illegitimate Packet Injection

In MUQAMI+, nodes are authenticated when they are deployed. Also, MUQAMI+ can be used in conjunction with state-of-the-art integrity checking mechanisms to ensure data integrity. Under such circumstances, it is not possible for adversary to inject false packets in the network. Existing security architectures [52], with which key management schemes can be coupled with, takes care of ”packet replaying” using methods such as initialization vectors. Also, there are simple methods to take care of attacks like ”hello flood” or ”injection of large number of packets”. Following are the attacks that can take place by injecting illegitimate packets and details of how MUQAMI+ provides defense against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.1.2 in chapter 2) Adversary can inject false application/routing information in the network to cause application malfunction e.g. Acknowledgement Spoofing.

Counter Measure: In MUQAMI+, all nodes are authenticated using pre-loaded authentication codes when they are deployed. For example, CH nodes send their authentication codes, for verification, to the CN in their discovery messages as follows:

-

$$m1 : \forall CH^i \in \{CH\} : CH^i \rightarrow CN : E_{K^i_{disc}}\{ID|Auth\_Code\}$$

Likewise, all sensor nodes send their pre-loaded authentication codes to their respective CH nodes in their discovery messages. then CH nodes authenticate them from the CN. After that, keys are distributed among only authentic sensor nodes and all communications are encrypted in secret keys, which can not be deciphered by outsiders. Also, outsiders can not encrypt messages using those secret keys to inject false information or spoof acknowledgement.

Attack 2: (See enumeration 2 in section 2.3.1.2 in chapter 2) Adversary can inject altered data/routing packets in the network.

Counter Measure: In order to inject altered data/routing packets in the network, adversary must be able to decipher packets that are being transferred in the network. Since all nodes are authenticated using pre-loaded authentication codes and secret keys are distributed among only authenticated nodes, adversary can not decipher messages exchanged between authenticated nodes. Therefore, adversary can not inject altered packets in the network.

Attack 3: (See enumeration 3 in section 2.3.1.2 in chapter 2) Adversary can access confidential information and pass it to an enemy.

Counter Measure: In order to access confidential information, adversary must be able to decipher packets that are being transferred in the network. Since all nodes are authenticated using pre-loaded authentication codes and secret keys are distributed among only authenticated nodes, adversary can not decipher messages exchanged between authenticated nodes. In MUQAMI+, all application related information is encrypted

using communication keys. Messages that report malicious activity are encrypted using pair-wise keys between sensor nodes and cluster head node and messages related to the refreshment of communication keys are encrypted in administrative keys or pair-wise keys. Therefore, an adversary can not access confidential information unless it knows a key.

Attack 4: (See enumeration 4 in section 2.3.1.2 in chapter 2) Adversary can inject large number of packets in the network to cause node outage or denial-of-service.
Counter Measure: An adversary can continuously send a large number of bogus packets in a WSN. In MUQAMI+, only authenticated nodes know encryption keys and packets from unauthenticated nodes are ignored. Also, this type of attack need not be dealt by key management schemes. Methods, other than key management, exist that can detect such attacks and then corrective measures can be taken. For instance, there is a limit to the amount of traffic that can be sent in WSN as WSN have a very limited communication bandwidth. Also, many sensor nodes are programmed to sleep at regular time intervals. Network can be programmed such that if the traffic exceeds certain threshold, attack is detected and communication is resumed on some other communication channel.

Attack 5: (See enumeration 5 in section 2.3.1.2 in chapter 2) Adversary can modify application/routing information to affect WSN operation.
Counter Measure: If an adversary can decipher application routing information, being transferred in the network, it can modify the information, create a new packet from it and send it towards the CN either pretending to be from a legitimate node or just as an outsider. If modified information is sent as an outsider, it will be ignored. If

information is sent as being from a legitimate node, the adversary requires proper decryption/encryption keys to decipher and then encipher the information so that other nodes accept it. However, encryption keys are not known to outsiders, which makes it impossible for adversary to send modified information in the network.

Attack 6: (See enumeration 6 in section 2.3.1.2 in chapter 2) Adversary can replay packets to cause node outage or routing issues.
Counter Measure: Efficient methods, other than those that involve key management, exist to take care of replaying attacks. For instance, replaying packets can be thwarted using initialization vectors as mentioned in [52]. Under such circumstances, it is not wise to use sensor nodes' precious energy to take care of packet replaying using key management.

Attack 7: (See enumeration 7 in section 2.3.1.2 in chapter 2) Adversary can send routing protocol's HELLO packets with more signal strength to cause a hello flood attack.
Counter Measure: In MUQAMI+, packets from only authenticated sources, among whom keys are distributed, are entertained. If an adversary tries to carry out HELLO flood attack, it would not be successful because member nodes would just ignore packets from unauthenticated source. If adversary keeps sending HELLO packets, it can only do it up to a certain limit because WSN have very limited communication bandwidth. If network traffic exceeds certain threshold, sensor nodes can signal malicious activity and nodes can shift to some other communication channel. If adversary is trying to create a wormhole, it would need to introduce a false node or compromise a node.

Table 5.2: Comparison of Defense Against Attacks due to Illegitimate Packets in WSN

	SHELL	**LEAP+**	**MUQAMI+**
Breach in Confidentiality	Yes	Yes	Yes
False Packet	Yes	Yes	Yes
Altered Packet	Yes	Yes	Yes
Packet Replaying	No	No	No
Large Amount of Packets	No	No	No
Hello Flood	Yes	Yes	Yes
Traffic Analysis	Yes	Yes	Yes

Attack 8: (See enumeration 8 in section 2.3.1.2 in chapter 2) Adversary can analyze traffic to determine nodes' responsibilities.

Counter Measure: It is possible for an adversary to probe certain nodes and monitor their responses to determine their responsibilities. In MUQAMI+, communication from unauthorized nodes are ignored and only authorized nodes know secret keys, which are used for communicating with other nodes. Still, it is possible that adversary can replay packets to probe sensor nodes. In that case, use of initialization vectors can thwart such attempts.

Like MUQAMI+, SHELL and LEAP+ schemes also employ mechanisms to authenticate the source when distributing keys. In these schemes also, unauthenticated source can not inject false packets, modify information in a packet or query member nodes for confidential information. Also, it is safe to assume for SHELL and LEAP+ that they do not respond to unauthenticated queries to give away confidential informa-

tion. In effect, SHELL and LEAP+ schemes provide same level of security against illegitimate packet injection as shown in Table 5.2.

5.1.3 Illegitimate Node Introduction

As already discussed above, MUQAMI+ can be used with state-of-the-art authentication and integrity checking mechanisms. Therefore, illegitimate node can neither access, modify, suppress or inject information in the network nor attract other nodes to route through it or spoof acknowledgements. Apart from that, illegitimate node can not carry out ”sybil attack” or ”wormhole attack”. As discussed previously, simpler methods exist, which take care of ”packet replaying”, ”hello flood” or ”injection of large number of packets” in the network. Following are the attacks that can occur because of the introduction of illegitimate node and details of how MUQAMI+ key management scheme for WSN defends against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.1.3 in chapter 2) Illegitimate node or false node can inject false information in the network.

Counter Measure: In MUQAMI+, all nodes are authenticated using pre-loaded authentication codes when they are deployed. For example, CH nodes send their authentication codes, for verification, to the CN in their discovery messages as follows: -

$$m1 : \forall CH^i \in \{CH\} : CH^i \rightarrow CN : E_{K^i_{disc}}\{ID|Auth\_Code\}$$

Likewise, all sensor nodes send their pre-loaded authentication codes to their respective CH nodes in their discovery messages. then CH nodes authenticate them from the CN (Refer to section 3.2.1 in chapter 3 for details). After that, keys are distributed among only authentic sensor nodes and all communications are encrypted in secret keys, which

can not be deciphered by unauthorized nodes. Also, outsiders can not encrypt messages using those secret keys to inject false information or spoof acknowledgement.

Attack 2: (See enumeration 2 in section 2.3.1.3 in chapter 2) Illegitimate node can inject a large number of packets in the network. If such packets are entertained by other nodes, it drains their energy. Otherwise it causes denial-of-service attack.
Counter Measure: In MUQAMI+, only authenticated nodes know encryption keys and packets from unauthenticated nodes are ignored. Moreover, more efficient methods that do not include key management exist to take care of such attacks. For instance, there is a limit to the amount of traffic that can be sent in WSN as WSN have a very limited communication bandwidth. Also, many sensor nodes are programmed to sleep at regular time intervals. Network can be programmed such that if the traffic exceeds certain threshold, attack is detected and communication is resumed on some other communication channel.

Attack 3: (See enumeration 3 in section 2.3.1.3 in chapter 2) Illegitimate node can breach information confidentiality by passing private information to a foe.
Counter Measure: In order to maintain confidentiality of private information, MUQAMI+ scheme uses secret keys for all communications. Every node is authenticated using pre-loaded values when it is deployed and secret keys are shared with only authenticated nodes. Therefore, an illegitimate node can not decipher confidential information concealed using secret keys.

Attack 4: (See enumeration 4 in section 2.3.1.3 in chapter 2) Illegitimate node can modify information being transferred from source node to sink node especially

during in-network processing. Also, it can cause problems in routing.

Counter Measure: IN MUQAMI+, sensor nodes are authenticated whenever they are deployed. After authentication, pair-wise keys are set up and then administrative keys are assigned to the sensor nodes. Communication keys are distributed in the end. An illegitimate nodes needs to get hold of a correct authentication code before it can decipher and encipher any information being sent to sink node and modify it. In MUQAMI+, it is not possible because authentication codes are initially sent using pre-loaded secret keys. Also, authentication codes can not be reused because they are unique for each sensor node.

Attack 5: (See enumeration 5 in section 2.3.1.3 in chapter 2) Illegitimate node can suppress information being transferred from one node to another to cause application malfunction or routing issues e.g. Selective Forwarding.

Counter Measure: It is possible for an illegitimate node to suppress information being sent through it towards the sink node. In order to do that, the node must be a part of the target network. In MUQAMI+, nodes are only allowed to join the network after they are authenticated by the command node. Also, nodes must present pre-loaded secret authentication codes, encrypted in pre-loaded pair-wise secret keys to join the network. Moreover, an authentication code can not be used more than once. Under these circumstances, it is not possible for an outsider node to act as a legitimate node and attract information.

Attack 6: (See enumeration 6 in section 2.3.1.3 in chapter 2) Illegitimate nodes can replay packets that have already been transmitted from one node to another. This can drain other sensor nodes' energy or create routing issues in the network.

Counter Measure: As already discussed above, efficient methods, other than those that involve key management, exist to take care of replaying attacks. For instance, replaying packets can be thwarted using initialization vectors as mentioned in [52]. Under such circumstances, it is unwise to direct sensor nodes' energies towards such an issue, which can be dealt with more efficiently, using simpler methods.

Attack 7: (See enumeration 7 in section 2.3.1.3 in chapter 2) Illegitimate node can attract other nodes to route their packets through it (to cause a sinkhole) so that it can modify/suppress information.

Counter Measure: In order to attract other nodes to route packets through it, an illegitimate node must be a part of the target network. If MUQAMI+ is applied to a WSN, nodes are only allowed to join the network after they are authenticated by the command node. Also, nodes must present pre-loaded secret authentication codes, encrypted in pre-loaded pair-wise secret keys to join the network. Moreover, an authentication node can not be used more than once. Under these circumstances, it is not possible for an outsider node to act as a legitimate node and attract other nodes to route their packets through it.

Attack 8: (See enumeration 8 in section 2.3.1.3 in chapter 2) Illegitimate node can spoof acknowledgements of dead or absent nodes.

Counter Measure: In order to spoof acknowledgement of a dead or an absent node, the illegitimate node must have secret keys that the dead or absent node shares with other nodes. Illegitimate node can not get secret keys until it sends a valid authentication code, which comes pre-loaded in sensor nodes and sent initially using pre-loaded keys. Also, authentication codes can only be used once. Therefore, an illegit-

Table 5.3: Comparison of Defense Against Attacks due to Illegitimate Nodes in WSN

	SHELL	**LEAP+**	**MUQAMI+**
Breach in Confidentiality	Yes	Yes	Yes
False Packet	Yes	Yes	Yes
Altered Packet	Yes	Yes	Yes
Packet Replaying	No	No	No
Large Amount of Packets	No	No	No
Hello Flood	Yes	Yes	Yes
Wormhole	Yes	Yes	Yes
Information Suppression	Yes	Yes	Yes
Sinkhole	Yes	Yes	Yes
Sybil	Yes	Yes	Yes

imate node can not gain access to a secret key and thus can not spoof acknowledgement.

Attack 9: (See enumeration 9 in section 2.3.1.3 in chapter 2) Illegitimate node can present multiple identities in a WSN i.e. carry out a sybil attack.

Counter Measure: If MUQAMI+ scheme is used in a WSN, even a legitimate node can not present multiple identities to carry out a sybil attack. Upon deployment, every node is required to present its identity, along with pre-loaded secret authentication code. Identities and authentication codes are sent using pre-loaded secret keys. Also, authentication codes can not be reused. Secret keys are disbursed to only those nodes, which present valid authentication codes. First, CH nodes are deployed and

authenticated as follows: -

$$m1 : \forall CH^i \in \{CH\} : CH^i \rightarrow CN : E_{K^i_{disc}}\{ID|Auth\_Code\}$$

Then all sensor nodes are authenticated and given initial key values as follows: -

$$\begin{aligned}
&\forall CH^i \in \{CH\} \wedge \forall SN^j \in \{SN_{CH^i}\} : \\
&\quad m1 : SN^j \rightarrow CH^i : E_{K^j_{disc}}\{ID(SN^j)|Auth_Code\} \\
&\quad m2 : CH^i \rightarrow CN : E_{K^i_{cn,ch}}\{ID(SN^j)|Auth_Code\} \\
&\quad m3 : CN \rightarrow CH^i : E_{K^i_{cn,ch}}\{E_{K^j_{bsc}}\{K^j_{bsc_new} \\
&\qquad |K^j_{disc_new}|K^1_{admin}|K^2_{admin}|...|K^k_{admin}\}\} \\
&\quad m4 : CH^i \rightarrow SN^j : E_{K^j_{disc}}\{K^{i,j}_{ch,sn}|E_{K^j_{bsc}} \\
&\qquad \{K^j_{bsc_new}|K^j_{disc_new}|K^1_{admin}|K^2_{admin}|...|K^k_{admin}\}\}
\end{aligned}$$

Under such circumstances, sybil attack can not be carried out. For more details regarding authentication procedures, refer to section 3.2.1 in chapter 3.

Attack 10: (See enumeration 10 in section 2.3.1.3 in chapter 2) Two illegitimate nodes can be introduced in a WSN to cause a wormhole in the network.

Counter Measure: In MUQAMI+, defense against introduction of illegitimate nodes to carry out wormhole attack is to authenticate every node during initial deployment and then disburse secret keys and information to it. If unauthenticated node is not allowed to join the network, wormhole can not be created. For instance, CH nodes are authenticated and then loaded with confidential information and secret keys as follows:

-

$$m1 : \forall CH^i \in \{CH\} : CH^i \rightarrow CN : E_{K^i_{disc}}\{ID|Auth\_Code\}$$

$$m2 : \forall CH^i \in \{CH\} : CN \rightarrow CH^i : E_{K^i_{disc}}\{K^i_{cn,ch}|EBS\_Matrix$$

$$|\forall SN^j \in \{\{SN_{CH^i}\} \cup \{KG_{CH^i}\}\} : \{ID(SN^j)|K^j_{disc}\}\}\}$$

Likewise other nodes are first authenticated and then provided with secret keys. This gives no room for an illegitimate nodes to join the network and cause a wormhole. For further details, refer to section 3.2.1 in chapter 3.

Attack 11: (See enumeration 11 in section 2.3.1.3 in chapter 2) An illegitimate node can cause a hello flood attack by sending routing protocol's HELLO packet with more signal strength.

Counter Measure: In MUQAMI+, all communications from unauthorized nodes are ignored so members node would not forward HELLO packets from illegitimate node. If adversary sends large number of HELLO packets to flood the network, it can only do it up to a certain limit because WSN have very limited communication bandwidth. If network traffic exceeds certain threshold, sensor nodes can signal malicious activity and nodes can shift to some other communication channel. Moreover, adversary can not use hello flood attacks to create a wormhole because unauthorized nodes can not be introduced in WSN when MUQAMI+ scheme is used.

Although LEAP+ assume time period during initial deployment phase to be safe, we do not suppose that adversary tries to introduce illegitimate node during that time. Otherwise, LEAP+ can not defend against any attack occurring due introduction of illegitimate node. Apart from the initial time period, LEAP+ has effective authentication mechanism to authenticate valid nodes before disbursing keys to them. Also, SHELL

scheme distributes keys to only authenticated nodes and do not entertain messages from unauthenticated nodes. Comparison of the security of the three schemes against the introduction of an illegitimate node is shown in Table 5.3.

5.1.4 Node Capture/Compromise

Due to the fact that WSN work unattended, there is always a possibility of node capture. Apart from recovering from node compromise, it is important to ensure that important nodes are not captured. MUQAMI+ has effective node revocation mechanisms to evict a compromised node from the network. After eviction, the compromised node does not remain part of the network. Thus, it can no longer access private information and secret keys. Also, it can not inject, modify or suppress information in the network. Moreover, it can not attract other nodes, cause ”acknowledgement spoofing”, carry out ”sybil” or ”wormhole” attack. However, EBS-based keys in MUQAMI+ are susceptible to node collusion attacks just like other EBS-based key management schemes. In case of collusion attack, MUQAMI+ reverts to pair-wise keys for security. Collusion prevention in EBS-based key management schemes is discussed in detail in [49]. In MUQAMI+, key management responsibility and responsibility of being cluster head node can be regularly transferred from one node to another. Therefore, no single node is of high importance than others. This helps in avoiding traffic analysis by an adversary. Also, this helps in avoiding to have high-valued targets in the network. Following are the attacks that can take place due to node compromise and details of how MUQAMI+ defends against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.1.4 in chapter 2) Adversary can access private information stored on the compromised sensor node and deliver it to a foe.

Counter Measure: Defense against node compromise consists of compromised node detection and compromised node revocation. MUQAMI+ scheme for WSN supports detection of malicious activity from a compromised node by having pair-wise keys between sensor nodes and CH nodes. However, complete procedure for compromised node detection is out of the scope of key management. Once a compromised node is detected, MUQAMI+ has highly effective mechanisms to revoke compromised nodes. For example, if CH node of cluster i is compromised, newly appointed/elected CH node is deployed as follows: -

$$m1 : CN \rightarrow CH^i : E_{K^i_{bsc}}\{EBS|\forall l \in \{\{SN_{CH^i}\} \cup \{KG_{CH^i}\}\} :$$
$$\{K^l_{disc}|E_{K^l_{bsc}}\{K^l_{bsc\_new}|K^l_{disc\_new}|CH\_Valid\}\}\}$$
$$m2 : \forall SN^k \in \{SN_{CH^i}\} : CH^i \rightarrow SN^k : E_{K^k_{disc}}$$
$$\{K^{i,k}_{ch,sn}|E_{K^k_{bsc}}\{K^k_{bsc\_new}|K^k_{disc\_new}|CH\_Valid\}\}$$
$$m2 : \forall KG^j \in \{KG_{CH^i}\} : CH^i \rightarrow KG^j : E_{K^j_{disc}}$$
$$\{K^{i,j}_{ch,kg}|E_{K^j_{bsc}}\{K^j_{bsc\_new}|K^j_{disc\_new}|CH\_Valid\}\}$$

Note that new CH node is given keys in such a manner that compromised CH node does not get to know any key values used after revocation. See section 3.2.3 for complete details on node revocation. After revocation, a compromised node can neither access private information nor access private keys in the network.

Attack 2: (See enumeration 2 in section 2.3.1.4 in chapter 2) Adversary can inject false application/routing information in the network. This can cause application malfunction or routing problems.

Counter Measure: If an adversary compromises a node in a WSN, it can use it to inject false information in the network. However, if a compromised node is detected,

MUQAMI+ has effective mechanisms to revoke such nodes from the network. For example, if a sensor node is compromised in cluster head i, it is revoked from the network as follows in section 3.2.3.2: -

$$m1 : \forall p \in K : CH^i \rightarrow KG^p : E_{K^{i,p}_{ch,kg}}\{Revoc\_Msg\}$$

$$m2 : \forall p \in K : KG^p \rightarrow CH^i : E_{K^{i,p}_{ch,kg}}\{E_{K^p_{admin}}\{K^p_{admin\_new}\}\}$$

$$m3 : \forall q \in M : CH^i \rightarrow KG^q : E_{K^{i,q}_{ch,kg}}\{\forall p \in K : E_{K^p_{admin}}\{K^p_{admin\_new}\}\}$$

$$m4 : \forall q \in M : KG^q \rightarrow * : E_{K^q_{admin}}\{\forall p \in K : E_{K^p_{admin}}\{K^p_{admin\_new}\}\}$$

Note that all administrative keys, which are known to the compromised node, are refreshed using administrative keys that are not known to it. After that, all further communication from the compromised node is ignored and thus adversary can not inject false information in the network.

Attack 3: (See enumeration 3 in section 2.3.1.4 in chapter 2) Adversary can access secret keys stored on the compromised node. Adversary can pass secret key to an illegitimate node or use it to query other legitimate nodes in the network.

Counter Measure: As discussed previously, MUQAMI+ has effective node revocation mechanisms after the system has detected a compromised node. MUQAMI+ uses EBS-based keys, not known to the compromised node, to refresh EBS-keys, known to the compromised node, such that after node revocation procedure is complete, evicted node does not know any key being used in the network. Pair-wise keys are unique for each node, so any further communication, which is done using pair-wise keys of the revoked node, is ignored. After that, even if revoked secret keys are used for sending packets, such packets are ignored.

Attack 4: (See enumeration 4 in section 2.3.1.4 in chapter 2) Adversary can turn off the compromised node so that it can no longer take part in the task assigned to the network.

Counter Measure: Physical capture of sensor nodes and turning them off is out of the scope of key management. There are other ways to protect against such attacks. For instance, a larger number of low cost sensor nodes can be used instead of a smaller number of more capable nodes under such circumstances to make it difficult for adversary to find every node and turn it off. If an adversary turns off the node after compromising it, it is less harmful than the case, in which it tries to use compromised node for accessing private information or causing system malfunction.

Attack 5: (See enumeration 5 in section 2.3.1.4 in chapter 2) Adversary can inject huge amount of traffic in the network to cause node outage or denial-of-service.

Counter Measure: In WSN, there is a limit to the amount of packets that can be sent because of the limited communication bandwidth in WSN. More efficient methods, other than the ones that involve key management, exist to thwart injection of huge amount of traffic in the network. For example, if traffic on a certain channel exceeds a certain threshold and most of it comes from a single node, communication can be shifted to some other communication channel and compromised node can be evicted from the network.

Attack 6: (See enumeration 6 in section 2.3.1.4 in chapter 2) Adversary can modify information to cause application malfunction or routing problems in the network.

Counter Measure: Once a compromised node is detected and revoked immediately, it no longer has access to confidential information or secret keys used in the network.

Thus, if an efficient compromised node detection mechanisms is in place, MUQAMI+ ensures that adversary can not access confidential information, modify it and send it towards the CN to cause application malfunction or routing problems.

Attack 7: (See enumeration 7 in section 2.3.1.4 in chapter 2) Adversary can suppress routing or application information being sent from one node to another.
Counter Measure: If an adversary wants to suppress information being sent from one node to another using a compromised node, it is mandatory that other nodes send information through the compromised node. If effective compromised node detection mechanism procedure is in place, nodes in the the network can be notified immediately about the compromised node and other nodes can stop sending packets through it. If MUQAMI+ is used, its effective node revocation mechanisms can evict the compromised node and protect from other hazards caused by a compromised node.

Attack 8: (See enumeration 8 in section 2.3.1.4 in chapter 2) Adversary can replay application/routing information to cause node outage or routing problems.
Counter Measure: In order to guard from such an attack, WSN must have effective compromised node detection mechanisms so that minimum damage is caused before MUQAMI+ key management scheme for WSN revokes the compromised node from the network. In fact, initialization vectors, as discussed in [52], can be used to detect packet replay attack and MUQAMI+ scheme can be used to revoke the compromised node, if an insider node is responsible for the attack.

Attack 9: (See enumeration 9 in section 2.3.1.4 in chapter 2) Compromised node can be used to attract nodes for routing their packets through it so that adversary can

modify/suppress information.

Counter Measure: In presence of an effective compromised node detection mechanism, which is assumed to be present in the WSN that employs MUQAMI+, it is not possible for a compromised node to attract traffic towards it for too long. After node detection, MUQAMI+ revokes the compromised node with an immediate effect so that the revoked node does not remain part of the network and can not attract other nodes to route their packets through it.

Attack 10: (See enumeration 10 in section 2.3.1.4 in chapter 2) Adversary can spoof acknowledgements of nodes, which are not present.

Counter Measure: In MUQAMI+, pair-wise keys are used for reporting of malicious activities. If malicious activity is witnessed by a node, it can report it immediately. Also, group keys are used for administrative and communication purposes and sensor nodes in WSN also have sense of direction [111]. Under such circumstances, it is not difficult for a compromised node detection mechanism to detect a compromised node, especially if it is spoofing acknowledgements for other nodes. Once a node compromise is detected, effective node revocation mechanisms of MUQAMI+ can be used to to revoke the compromised node.

Attack 11: (See enumeration 11 in section 2.3.1.4 in chapter 2) Adversary can use a compromised node to cause a sybil attack.

Counter Measure: In MUQAMI+, nodes are authenticated by the CN using authentication codes as soon as they are deployed. Also, authentication code for each node is unique and can not be used more than once. In order to register multiple identities, a malicious node must be able to produce valid identities and authentication code or steal

an identity and an authentication code, which is not possible because authentication codes are sent to CN in discovery messages and can not be reused. If a compromised node presents identity of some other node or a dead node, it can lead to its detection as other nodes can overhear messages, have sense of direction and can report malicious activities using pair-wise keys.

Attack 12: (See enumeration 12 in section 2.3.1.4 in chapter 2) Compromised node can be used to carry out hello flood attack.
Counter Measure: As discussed previously, WSN have very limited communication bandwidth, which limits the number of packets that an adversary can inject in the network. Hello flood attack can easily be detected by monitoring amount of traffic in the network and from each node. If traffic exceeds certain threshold, attack can be reported. After an attack has been detected, effective node revocation procedures of MUQAMI+ can be used to evict compromised node from the network.

Attack 13: (See enumeration 13 in section 2.3.1.4 in chapter 2) Compromised node can collude with another compromised node or an illegitimate node to cause wormhole in the target network.
Counter Measure: If two insider nodes, or an insider and an outsider node, collude using an out of band communication channel to create a wormhole, it can cause maximum damage to EBS-based key management schemes. If an insider and an outsider node collude and use in-band communication channel, communication between compromised node and an outsider node can lead to compromised node detection because other sensor nodes can overhear messages using group keys, sensor nodes have sense of direction and too much private communication between two sensor nodes lead to suspicion. If

Table 5.4: Comparison of Defense Against Attacks due to Node Compromise in WSN

	SHELL	**LEAP+**	**MUQAMI+**
Breach in Confidentiality	Yes	Yes	Yes
False Packet	Yes	Yes	Yes
Altered Packet	Yes	Yes	Yes
Packet Replaying	No	No	No
Large Amount of Packets	No	No	No
Hello Flood	Yes	Yes	Yes
Wormhole	Yes	Yes	Yes
Information Suppression	Yes	Yes	Yes
Sinkhole	Yes	Yes	Yes
Sybil	Yes	Yes	Yes
Key Exposition	Yes	Yes	Yes
Node Removal	No	No	No

they use out-of-band communication channel, other nodes can use message overhearing to detect that messages are not being sent forward. In both cases, compromised node can be detected and the node revocation procedures of MUQAMI+ can be used to evict the insider node. Without having an insider node on its side, outsider node can not continue the attack. However, if two or more insider nodes are compromised, there is a possibility that number of compromised administrative keys increase up to the level, after which EBS matrix can not be used to revoke compromised nodes. Under such circumstances, MUQAMI+ scheme reverts back to pair-wise keys for revoking compromised nodes.

All state-of-the-art key management schemes for WSN assume presence of an efficient compromised node detection mechanism because detection of compromised nodes is out of scope of key management schemes for WSN. SHELL and LEAP+ schemes have efficient node revocation mechanisms, which guard from all attacks mentioned before and exposition of secret keys, which occurs because of node compromise. Like MUQAMI+, SHELL and LEAP+ do not provide defense against physical attacks such as node removal. Comparison of defense of the three schemes, in presence of efficient node revocation mechanism, is shown in Table 5.4.

5.1.5 Communication Disruption

Communication disruption by creating noise in the channel or jamming radio communication occurs at the physical layer. Therefore it is out of scope of key management schemes to guard against such attacks. For countermeasures against jamming attacks, refer to [112], [113]. Following are the attacks that can take place at physical layer and suggestions regarding defense against these atacks: -

Table 5.5: Comparison of Defense Against Communication Disruption Attacks in WSN

	SHELL	**LEAP+**	**MUQAMI+**
Signal Jamming	No	No	No
Channel Noise	No	No	No

Attack 1: (See enumeration 1 in section 2.3.1.5 in chapter 2) Adversary can jam radio signal, through which sensor nodes communicate e.g. Denial of Service Attack.
Counter Measure: Jamming of radio signal is an attack that occur on physical layer and thus it is out of scope of key management scheme for WSN.

Attack 2: (See enumeration 2 in section 2.3.1.5 in chapter 2) Adversary can introduce a lot of noise in the channel, through which sensor nodes communicate.
Counter Measure: Introducing a lot of noise in communication channel is also out of scope of key management scheme for WSN as this attack also occurs on physical layer of WSN.

As attacks related to disruption of communication are out of scope of key management schemes for WSN, other state-of-the-art key management schemes for WSN also do not provide defense against such attacks as shown in Table 5.5.

5.2 BARI+ (Wireless Body Area Networks)

Just like MUQAMI+, we have four types of keys in BARI+ scheme for wireless body area networks: communication key, administrative keys, pair-wise keys between sensor

nodes and the medical server (MS) and pair-wise keys between sensor nodes and personal server node (PS) node. In this case too, each type of key has a significance in maintaining security and ensuring that the target network works normally.

Key management scheme for WBAN should not waste energy of sensor nodes in protecting against attacks that involve routing. In WBAN, Personal Server (PS) is in direct communication range of many nodes. Nodes, which have very limited communication range, normally select one nearby node to relay their information to the PS rather than selecting paths. Therefore, WBAN is not much vulnerable to routing attacks such as ”selective forwarding”, ”sinkhole”, ”sybil”, ”wormhole” and attacks, in which routing information is spoofed, altered or replayed. In WBAN, not all sensor nodes sense the same phenomena as in WSN. Still, it is more efficient to use single group key for the network as refreshing all pair-wise keys incurs more overhead than refreshing a single group key.

As in the case of WSN, repetitive use of a single group key (communication key) for communication makes the key more vulnerable to cryptanalytic attacks and compromise of the key or a single node compromises the whole cluster. In order to avoid cryptanalytic attacks, the single group key must be refreshed regularly. Also, it should be refreshed using some other key to maintain forward secrecy. If single group key is refreshed using pair-wise keys, it is not energy efficient because the cluster head node will have to send separate message to each node in its cluster for key refreshment. It is more efficient to use other group keys (administrative keys) for refreshing the single group key. As the other group keys are used rarely, they are less vulnerable to cryptanalytic attacks and can be used to refresh themselves for some time, after which they must be refreshed using pair-wise keys between sensor nodes and the PS node. Moreover, sensor nodes must share pair-wise keys with the MS to ensure secure initial deployment and secure

replacement of a compromised PS node.

It is the use of group keys for key refreshment in a distributed manner and use of biometric values for key generation that makes BARI+ more efficient than other schemes. In this section, I present security analysis of BARI+ scheme with respect to vulnerabilities and attack vectors discussed in chapter 2. Under each vulnerability, each attack vector and effectiveness of BARI+ under that vector is discussed. Also, we compare BARI+ with related key management schemes in terms of security.

5.2.1 Passive Listening

In BARI+, all message exchanges are secured using secret keys. A passive adversary, listening to communications, can not comprehend messages unless it obtains secret keys. However, a passive adversary can carry out cryptanalytic attacks on secret keys. To avoid cryptanalytic attacks, keys are refreshed at regular intervals. Following are the attacks that can occur because of passive listening and details of how BARI+ defends against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.2.1 in chapter 2) Adversary can listen to the private communication between two sensor nodes so that confidentiality of information is breached.

Counter Measure: In BARI+, all communications are encrypted in secret keys starting from initial deployment phase. When deployed, all nodes are pre-loaded with key values, which they use to encrypt discovery messages. Encryption of confidential messages using secret keys continue throughout the network lifetime. For example, a SN node communicates application data to its PS in the following manner: -

$$message : SN \rightarrow PS : E_{K_{comm}}\{Application\_Data\}$$

Refer to Section 4.2 in chapter 4 to see how each message exchanged in BARI+ scheme is encrypted using secret keys.

Attack 2: (See enumeration 2 in section 2.3.2.1 in chapter 2) Adversary can carry out cryptanalytic attacks to reveal secret keys.
Counter Measure: To prevent from cryptanalytic attacks, all secret keys are regularly refreshed. Communication key is refreshed using administrative keys and pair-wise keys are used to refresh administrative keys periodically to maintain forward secrecy. For example, following message exchanges take place when PS node refreshes communication key using current value of administrative key: -

$$m1 : PS \rightarrow * : E_{K_{admin}}\{K_{comm}|Auth\_Code^{PS}\}$$

For complete detail on key refreshment procedures of BARI+, refer to section 4.2.2 in chapter 4.

LEAP+ scheme has the capability of avoiding single-point-of-failure and MUQAMI+ has the capability of avoiding single-point-of-failure and distributing responsibilities among multiple nodes. However, architecture of WBAN is such that all traffic is sent towards a single node i.e. the PS (Personal Server) node and only the PS has responsibility of processing the data and/or forwarding it to the MS (Medical Server) node. Therefore, exposition of single-point-of-failure and traffic analysis, to discover an important node, is not applicable to WBAN. MUQAMI+ and LEAP+ both provide protection against cryptanalytic attacks and breach in confidentiality as shown in Table 5.6.

Table 5.6: Comparison of Defense Against Attacks due to Passive Listening in WBAN

	MUQAMI+	**LEAP+**	**BARI+**
Breach in Confidentiality	Yes	Yes	Yes
Cryptanalysis	Yes	Yes	Yes

5.2.2 Illegitimate Packet Injection

In order to provide authentication, BARI+ uses authentication codes in all communications. Also, it provides mechanisms to refresh them. In this way, all receiving nodes know origins of a message. If a message does not have a valid authentication code, it is discarded and malicious activity is indicated. If an illegitimate node sends a message, containing authentication code of a legitimate node, the legitimate node can indicate malicious activity. Also, BARI+ can be used with state-of-the-art integrity checking mechanisms. Under such circumstances, it is not possible for adversary to inject false packets in the network. "Packet replaying" can be curbed using methods such as initialization vectors. Following are the attacks that can take place due to injection of illegitimate packets in a WBAN and details of how BARI+ defends against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.2.2 in chapter 2) Adversary can inject false application information or altered data packets in the network to cause application malfunction.

Counter Measure: In BARI+, nodes are authenticated not only in the initial deployment phase but also whenever they send an important message. For example, when a sensor (SN) node i refreshes administrative key on its turn according to the key

refreshment schedule, it has to present its valid authentication code as follows: -

$$m1 : SN^i \rightarrow * : E_{K^{old}_{admin}}\{K^{new}_{admin}|Auth\_Code^i\}$$

Apart from that, false application information can not be injected through an illegitimate packet because secret keys are distributed among authenticated nodes only.

Attack 2: (See enumeration 2 in section 2.3.2.2 in chapter 2) Adversary can access confidential information and pass it to an enemy.
Counter Measure: In order to access confidential information, adversary needs secret keys. In BARI+ scheme for WBAN, secret keys are distributed among only the authenticated nodes. Nodes are authenticated by examining authentication codes pre-loaded in them. When deployed initially, nodes present their identities and authentication codes, encrypted in pre-loaded key values so that an outsider can not get to know authentication codes. Also, an outsider can not use an authentication code that is already in use of some insider node. Apart from that, authentication codes are refreshed at regular intervals to avoid their theft. Under such circumstances, it is not possible for an adversary to inject false packet and probe sensor nodes for confidential information.

Attack 3: (See enumeration 3 in section 2.3.2.2 in chapter 2) Adversary can inject large number of packets in the network to cause node outage or denial-of-service.
Counter Measure: In BARI+ key management scheme for WBAN, all packets form unauthorized sources are ignored and nodes do not spend their energies in deciphering and processing packets from unauthorized nodes. Still, an adversary can keep sending packets on communication link to cause denial-of-service. Just like WSN, communication bandwidth of WBAN is very limited and thus adversary can only inject packets up to a certain limit. Detection of an attack is easier in WBAN than WSN. In addition

Table 5.7: Comparison of Defense Against Attacks due to Illegitimate Packets in WBAN

	MUQAMI+	**LEAP+**	**BARI+**
Breach in Confidentiality	Yes	Yes	Yes
False Packet	Yes	Yes	Yes
Altered Packet	Yes	Yes	Yes
Packet Replaying	No	No	Yes
Large Amount of Packets	No	No	No

to programming the network to report malicious activity if network traffic exceeds certain limit, PS node can easily figure out source of the problem since all nodes, which can cause such an attack, are in communication range of each other. Apart from that, WBAN are deployed in environments, which can be physically cleared from sources that are involved in such activities.

Attack 4: (See enumeration 4 in section 2.3.2.2 in chapter 2) Adversary can modify application information to affect WBAN operation.

Counter Measure: An adversary can listen to communication in WBAN, modify it and then try to inject in WBAN. In order to modify an information to something meaningful, adversary must be able to decipher secret information using secret keys, which are only known to authenticated nodes. Even if an adversary can decipher secret information, it can not inject modified information in the network because all communication from unauthorized sources are ignored in WBAN, which employ BARI+ key management scheme.

Attack 5: (See enumeration 5 in section 2.3.2.2 in chapter 2) Adversary can replay packets to cause node outage.

Counter Measure: In BARI+, communications from unauthenticated sources are ignored unless the network is in initialization phase or node addition phase. In WBAN, it is easy to detect a replayed packet because all nodes are in communication range of each other and the node, whose packet is replayed, can easily detect the attack. Even if the network is in initial deployment or node addition phase, MS sends identities and authentication codes of nodes, which are to join the network, to the PS node. If packets are replayed during these phases, PS node can easily detect malicious activity. Since WBAN operate in an environment, in which human intervention is possible, source of such malicious activity can be removed physically from the environment.

Attacks that can occur in WBAN due to the introduction of illegitimate packets, do not include attacks that include routing e.g. HELLO flood attacks as most of the nodes in WBAN have the PS in their communication range. As discussed in the previous section (5.1), MUQAMI+ and LEAP+ schemes do not entertain packets from unauthorized sources so that adversary can not inject false or modified information in the network or breach confidentiality. Also, LEAP+ and MUQAMI+ do not provide defense against packet replaying or introduction of huge traffic in WBAN. Comparison of LEAP+, MUQAMI+ and BARI+ schemes with regard to defense against illegitimate packets is shown in Table 5.7.

5.2.3 Illegitimate Node Introduction

As already discussed above, BARI+ uses and manages authentication codes in all communications. Also, wrong use of authentication codes can be easily detected by

relevant nodes. Moreover, BARI+ can be used with state-of-the-art mechanisms for integrity checking. Therefore, illegitimate node can not access, modify, suppress or inject information in the network. Following is the listing of attacks and that occur due to the introduction of illegitimate nodes and details of the way in which BARI+ scheme offers defense against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.2.3 in chapter 2) Illegitimate node or false node can inject false or altered information in the network.

Counter Measure: If BARI+ scheme is employed in WBAN, a false node can not join the network because while joining the network, every node has to present a valid identity and authentication code using a pre-loaded secret key. For example, discovery message is sent by every sensor node during initial deployment: -

$$m1 : \forall i \quad if \exists SN^i : SN^i \rightarrow PS : E_{K_{admin}}\{ID^i | Auth\_Code^i\}$$

If a node is not part of the network, it does not have access to secret keys to decipher and encipher secret information to be injected in the network. Even if a node gets access to a secret key, it can not inject information because all messages from unauthorized nodes are ignored. If an outsider node tries to use identity of a legitimate node, it can easily be detected because, in WBAN, most of the nodes are in communication range of each other and can overhear messages.

Attack 2: (See enumeration 2 in section 2.3.2.3 in chapter 2) Illegitimate node can inject a large number of packets in the network. If such packets are entertained by other nodes, it drains their energy. Otherwise it causes denial-of-service attack.

Counter Measure: In BARI+, all communications from unauthorized nodes are ignored and thus not entertained by legitimate nodes. However, outsider node can inject

large amount of traffic to cause denial-of-service attack. In WBAN scenario, it is very easy to track source of such an attack. If traffic from a particular source exceeds certain limits, attack can be indicated. As human intervention is possible in WBAN, source of the trouble can be easily singled out and removed from the environment.

Attack 3: (See enumeration 3 in section 2.3.2.3 in chapter 2) Illegitimate node can breach information confidentiality by passing private information to a foe.
Counter Measure: In order to pass private information to a foe, an illegitimate node must be able to decipher secret information. In order to decipher secret information, adversary must have secret key. In BARI+ scheme for WBAN, no unauthorized node is granted access to secret keys. For example, when a sensor node refreshes communication key on its turn, it uses current value of administrative key as follows: -

$$m1 : PS \rightarrow * : E_{K_{admin}}\{K_{comm}|Auth\_Code^{PS}\}$$

An outsider node can not access this private information because it does not know secret administrative key.

Attack 4: (See enumeration 4 in section 2.3.2.3 in chapter 2) Illegitimate node can suppress information being transferred from one node to another.
Counter Measure: In WBAN, it is possible that some nodes have a very limited communication range i.e. few centimeters and can not reach PS directly. These nodes send their packets through some other node, which is part of the network and can access PS directly. In order to suppress information from nodes, which have very limited communication range, an illegitimate node must be part of the network. An illegitimate node can not be part of the network unless it bears a valid identity and a valid authentication code. If an illegitimate node is not part of the network, neither it

Table 5.8: Comparison of Defense Against Attacks due to Illegitimate Nodes in WBAN

	MUQAMI+	**LEAP+**	**BARI+**
Breach in Confidentiality	Yes	Yes	Yes
False Packet	Yes	Yes	Yes
Altered Packet	Yes	Yes	Yes
Packet Replaying	No	No	Yes
Large Amount of Packets	No	No	No
Information Suppression	Yes	Yes	Yes

can decipher discovery messages from nodes, which have very limited communication range, nor it can authorization from PS to route packets from nodes, which have very limited communication range. So, illegitimate node can not sit between other sensor nodes and PS node to suppress information.

Attack 5: (See enumeration 5 in section 2.3.2.3 in chapter 2) Illegitimate nodes can replay packets that have already been transmitted from one node to another. This can drain other sensor nodes' energy.

Counter Measure: All messages from unauthorized nodes are ignored when BARI+ scheme is applied in WBAN. Also, all nodes are in communication range of each other and can overhear messages if they are replayed. If an illegitimate node replays a packet, the node, which originated the packet can easily detect malicious activity, which can be sorted out either physically or by shifting to some other communication channel. However, there are certain messages in BARI+, which can cause damage, if replayed. Timestamps are included in such messages so that the damage does not go too far before

source of problem is sorted out. For example, PS always includes timestamp when it issues a new key refreshment schedule as follows: -

$$m1 : PS \rightarrow * : E_{K_{admin}}\{Key\_Ref\_Schedule|Auth\_Code^{PS}|Timestamp\}$$

Including timestamp avoids confusion that may occur between sensor nodes regarding key refreshment schedule.

Just as in the case of BARI+, LEAP+ and MUQAMI+ schemes provide protection against the introduction of illegitimate nodes in a WBAN except when it replays a packet or injects huge amount of traffic in the network. Although LEAP+ scheme assumes initial safe time period in the network, this assumption is more realistic in WBAN as in case of WSN because of possible human intervention. Comparison of BARI+ with LEAP+ and MUQAMI+ schemes is shown in Table 5.8. Note that attacks, related to routing are not applicable in WBAN scenario.

5.2.4 Node Capture/Compromise

Although human intervention is possible in WBAN, it is not always possible to remove or turn off compromised node from the network. Therefore BARI+ key management scheme for WBAN provide efficient node revocation mechanism so that compromised node is prevented from accessing private information, modifying useful information, injecting useless information or replaying packets. In BARI+, key management responsibility is distributed among multiple sensor nodes. Therefore, WBAN can function normally even if a node is compromised. Following are the attacks that can take place due to node compromise and details of how BARI+ defends against these attacks: -

Attack 1: (See enumeration 1 in section 2.3.2.4 in chapter 2) Adversary can access private information stored on the compromised sensor node and deliver it to a foe.
Counter Measure: BARI+ key management scheme for WBAN has effective mechanism for revocation of compromised nodes from the network. Once a compromised node is detected, administrative and communication keys are refreshed using pair-wise keys as follows: -

$$m1 : \forall i \quad if \exists SN^i : PS \rightarrow SN^i : E_{K^i_{bsc\_old}}\{K_{admin}|K^i_{bsc\_new}$$
$$|Auth\_Code^i_{new}|Auth\_Code^{PS}\}$$
$$m2 : PS \rightarrow * : E_{K_{admin}}\{K_{comm}|Auth\_Code^{PS}\}$$

Above message is sent to all nodes except the compromised node. Note that authentication codes are also refreshed so that compromised node can not use them. After that, compromised node does not have access to secret keys to decipher confidential information. Detection of compromised nodes is out of the scope of key management in WBAN. WBAN span a small area, with most of the nodes within communication range of each other. This renders compromised node detection easier in WBAN as compared to WSN.

Attack 2: (See enumeration 2 in section 2.3.2.4 in chapter 2) Adversary can access secret keys stored on the compromised node. Adversary can pass secret key to an illegitimate node or use it to query other legitimate nodes in the network.
Counter Measure: If WBAN key management scheme is employed in WBAN, the network must have efficient compromised node detection mechanisms. As soon as a compromised node is detected, it is evicted from the network using pair-wise keys immediately. After being evicted from the network, communication and administrative keys are refreshed in such a way that new values are not know to the compromised node

and thus a compromised node can neither use it to query legitimate nodes nor pass it to an illegitimate node.

Attack 3: (See enumeration 3 in section 2.3.2.4 in chapter 2) Adversary can inject false or modified application information in the network. This can cause application malfunction.

Counter Measure: In order to inject false or modified application information in the network, compromised node must be able to encipher and decipher confidential information being sent in the network. If node compromise is detected effectively, then BARI+ scheme ensures that compromised node is evicted as soon as it is detected, after which it does not have access to new values of secret keys. After being evicted, compromised node can not inject false or modified information because it can not encipher and decipher information as it does not have access to secret keys.

Attack 4: (See enumeration 4 in section 2.3.2.4 in chapter 2) Adversary can turn off the compromised node so that it can no longer take part in the task assigned to the network.

Counter Measure: Turning off a sensor node falls under the category, in which physical assault is carried out on the target network. Dealing with such circumstances is out of scope of key management scheme for WBAN. In WBAN, it is fairly easy to deal with such circumstances as human intervention is possible and imposing physical security is also possible unlike WSN.

Attack 5: (See enumeration 5 in section 2.3.2.4 in chapter 2) Adversary can inject huge amount of traffic in the network to cause node outage or denial-of-service.

Counter Measure: Just like WSN, communication bandwidth of WBAN is very limited. Therefore, there is a limit to the amount of traffic that the compromised node can inject in the network. A fairly simple way to detect such an attack is to monitor amount of traffic in the network. If traffic exceeds certain threshold, source of the problem (compromised node) can be singled out. After that, if human intervention is possible, compromised node is removed from environment. If human intervention is not possible, then nodes can shift to some other communication channel and compromised node can be evicted using mechanisms introduced in BARI+ key management scheme for WBAN.

Attack 6: (See enumeration 6 in section 2.3.2.4 in chapter 2) Adversary can suppress information being sent from one node to another.

Counter Measure: In WBAN, it is possible that some nodes have a very limited communication range i.e. few centimeters and can not reach PS directly. These nodes send their packets through some other node, which is part of the network and can access PS directly. Also, all information from the network is sent through PS node. Although PS has more capabilities than sensor nodes, it is also a battery powered device. If a an intermediate legitimate node is compromised, it can be used to suppress information being sent from the WBAN to MS. Given that an effective compromised node detection mechanism is in place, BARI+ has effective mechanisms for evicting any type of node from the network. For example, if PS is compromised new PS node is deployed and

Table 5.9: Comparison of Defense Against Attacks due to Node Compromise in WBAN

	MUQAMI+	**LEAP+**	**BARI+**
Breach in Confidentiality	Yes	Yes	Yes
False Packet	Yes	Yes	Yes
Altered Packet	Yes	Yes	Yes
Packet Replaying	No	No	Yes
Large Amount of Packets	No	No	No
Information Suppression	Yes	Yes	Yes
Key Exposition	Yes	Yes	Yes
Node Removal	No	No	No

keys are established as follows: -

$$m1 : \forall i \quad if \exists SN^i : PS \rightarrow SN^i : E_{K^i_{SN,MS\_old}}\{K^i_{bsc} | Auth\_Code^{PS}_{new} | K^i_{SN,MS\_new} | Auth\_Code^{MS}\}$$

$$m2 : \forall i \quad if \exists SN^i : PS \rightarrow SN^i : E_{K^i_{bsc\_old}}\{K_{admin} | K^i_{bsc\_new} | Auth\_Code^i_{new} | Auth\_Code^{PS}\}$$

$$m3 : PS \rightarrow * : E_{K_{admin}}\{K_{comm} | Auth\_Code^{PS}\}$$

For details regarding compromised node eviction and key refreshment procedures of BARI+, refer to section 4.2.2 in chapter 4. After being evicted, a compromised node becomes an outsider and neither packets are routed through it nor it can decipher them.

Attack 7: (See enumeration 7 in section 2.3.2.4 in chapter 2) Adversary can replay application information to cause node outage.

Counter Measure: In WBAN, all nodes are in communication range of each other and can overhear messages if they are replayed. If a compromised node replays a packet, the node, which originated the packet can easily detect malicious activity, which can lead to compromised node detection. After a compromised node is detected, effective mechanisms of BARI+ can be used to evict compromised node. However, there are certain messages in BARI+, which can cause damage, if replayed. Timestamps are included in such messages so that the damage does not go too far before source of problem is sorted out. For example, PS always includes timestamp when it issues a new key refreshment schedule as follows: -

$$m1 : PS \rightarrow * : E_{K_{admin}}\{Key\_Ref\_Schedule|Auth\_Code^{PS}|Timestamp\}$$

Including timestamp avoids confusion that may occur between sensor nodes regarding key refreshment schedule.

As LEAP+ and MUQAMI+ schemes are designed for large scale WSN, they provide effective mechanisms for node revocation given effective mechanisms for compromised node detection exist. Apart from other attacks, node compromise can result in the removal of compromised node or keys on the compromised node can be exposed. Like BARI+, MUQAMI+ and LEAP+ have effective mechanisms to refresh keys in such a way that the compromised node does not get to know new key values. Defense against removal of a node is not applicable to key management schemes. Comparison of the three schemes in defense against attacks in WBAN that can occur due to node compromise, given effective mechanisms for compromised node detection exist, is shown in Table 5.9.

Table 5.10: Comparison of Defense Against Communication Disruption Attacks in WBAN

	MUQAMI+	**LEAP+**	**BARI+**
Signal Jamming	No	No	No
Channel Noise	No	No	No

5.2.5 Communication Disruption

Just like WSN, WBAN are also susceptible to attacks that occur at the physical layer at out of scope of key management schemes. Following are the attack that can take place at physical layer and suggestions regarding protection against such attacks: -

Attack 1: (See enumeration 1 in section 2.3.2.5 in chapter 2) Adversary can jam radio signal, through which sensor nodes communicate e.g. Denial of Service Attack.
Counter Measure: Jamming of radio signal is an attack that occur on physical layer and thus it is out of scope of key management scheme for WBAN. Besides, such attacks can be easily sorted out in WBAN environment because of possible human intervention.

Attack 2: (See enumeration 2 in section 2.3.2.4 in chapter 2) Adversary can introduce a lot of noise in the channel, through which sensor nodes communicate.
Counter Measure: Introducing a lot of noise in communication channel is also out of scope of key management scheme for WBAN as this attack also occurs on physical layer of WBAN. Also, possible human intervention in WBAN makes it easier to sort out source of such attacks in WBAN.

As attacks related to disruption of communication are out of scope of key management schemes for WBAN too, other key management schemes, with which we compare, also do not provide defense against such attacks as shown in Table 5.10.

Chapter 6

Conclusions and Future Directions

6.1 Conclusions

This thesis aims to achieve scalable, distributed and efficient key management solution for WSN by distributing key management responsibility locally within cluster and using EBS matrices. Its primary contribution lies in the development of unified, energy-efficient framework, called scalable and energy efficient key management framework. Main contributions of this thesis are summarized as follows: -

I present MUQAMI+ key management scheme for clustered wireless sensor networks. I compare MUQAMI+ with two other state of the art schemes and found that my scheme is more efficient in key refreshment and node revocation phases. MUQAMI+ uses EBS matrix to manage large number of nodes with small number of keys. MUQAMI+ avoids single points of failure in clustered WSN by distributing key management among multiple sensor nodes locally i.e. within a cluster. Also, MUQAMI+ has effective key revocation mechanisms. In addition to that, MUQAMI+ is flexible i.e. responsibilities of key management and being cluster head node can be shifted from one node to another.

Differences between WSN and WBAN in terms of network characteristics, application characteristics and security requirements are highlighted. Then I propose a key

management scheme BARI+ for WBAN that exploits the network and application characteristics of WBAN. Also, I provide analysis of my scheme and compare it with other schemes. BARI+ uses biometric values from WBAN applications to generate random key values. BARI+ distributes key management responsibility among multiple nodes, which are in communication range of each other, in the network. BARI+ does not require sensor nodes to sense more than one biometric from human body and it does not assume that the sensor nodes have perfect time synchronization.

6.2 Future Directions

Key management is important in defending against attacks on wireless sensor networks. Key management provides attack prevention mechanisms in wireless sensor networks. Also, it provides mechanisms to deal with attacks when they occur. However, attack detection is also a very important part of security. One can not deal with an attack if he can not detect it properly. Apart from that, it is important to focus on other security features of WSN such as privacy [114] and trust [111].

Future direction of this research is to focus on attack detection mechanisms for WSN and WBAN. Also, applications of wireless sensor networks should be critically analyzed separately when researching for their attack detection mechanisms. It may happen that it is easy to detect attacks in one application environment as compared to others. Also, future directions of this research aim towards focusing on other security features, such as privacy and trust, to provide more completeness in security of WSN.

Bibliography

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: A survey. *Computer Networks*, 38(4):393–422, 2002.

[2] S. Tilak, N.B. Abu-Ghazaleh, and W. Heinzelman. A taxonomy of wireless microsensor network models. *ACM SIGMOBILE Mobile Computing and Communications Review*, 6(2):28–36, April 2002.

[3] S.M.K. Raazi, Z. Pervez, and S. Lee. *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*, chapter [Key Management Schemes of Wireless Sensor Networks: A Survey]. Auerbach Publications, CRC Press, Taylor & Francis Group, USA., 2010.

[4] N. Xu. A survey of sensor network applications. *IEEE Communications Magazine*, 40:102–114, August 2002.

[5] V. Cantoni, L. Lombardi, and Lombardi P. Future scenarios of parallel computing: Distributed sensor networks. *Journal of Visual Languages & Computing*, 18(5):484 – 491, October 2007.

[6] R.M. Ruair, M.T. Keane, and G. Coleman. A wireless sensor network application requirements taxonomy. pages 209–216, Cap Esterel, France, August 25-31 2008. IEEE Computer Society.

[7] R. Szewczyk, J. Polastre, A. Mainwaring, and A. Culler. Lessons from a sensor network expedition. In *EWSN 2004: Proceedings of the 1st European Workshop on Sensor Networks*, volume 2920 of *Lecture Notes in Computer Science*, pages 307–322, Berlin, Germany, January 19-21 2004. Springer.

[8] A. Mainwaring, D. Culler, J. Polastre, R. Szewczyk, and J. Anderson. Wireless sensor networks for habitat monitoring. In *WSNA 2002: Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications*, pages 88–97, Atlanta, Georgia, USA, September 28 2002. ACM.

[9] R. Holman, J. Stanley, and T. Ozkan-Haller. Applying video sensor networks to nearshore environment monitoring. *IEEE Pervasive Computing*, 2(4):14–21, October 2003.

[10] K. Martinez, J.K. Hart, and R. Ong. Environmental sensor networks. *Computer*, 37(8):50–56, August 2004.

[11] D. Estrin, R. Govindan, J. Heidemann, and S. Kumar. Next century challenges: scalable coordination in sensor networks. In *MobiCom 1999: Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, pages 263–270, Seattle, Washington, USA, August 15-20 1999. ACM.

[12] D. Braginsky and D. Estrin. Rumor routing algorthim for sensor networks. In *WSNA 2002: Proceedings of the 1st ACM international workshop on Wireless*

sensor networks and applications, pages 22–31, Atlanta, Georgia, USA, September 28 2002. ACM.

[13] I.F. Akyildiz, D. Pompili, and T. Melodia. Underwater acoustic sensor networks: research challenges. *Ad Hoc Networks*, 3(3):257 – 279, 2005.

[14] C. Gui and P. Mohapatra. Power conservation and quality of surveillance in target tracking sensor networks. In *MobiCom 2004: Proceedings of the 10th annual international conference on Mobile computing and networking*, pages 129–143, Philadelphia, PA, USA, September 26 - October 1 2004. ACM.

[15] T. He, S. Krishnamurthy, J.A. Stankovic, T. Abdelzaher, L. Luo, R. Stoleru, T. Yan, L. Gu, J. Hui, and B. Krogh. Energy-efficient surveillance system using wireless sensor networks. In *MobiSys 2004: Proceedings of the 2nd international conference on Mobile systems, applications, and services*, pages 270–283, Boston, MA, USA, June 6-9 2004. ACM.

[16] T. Yan, T. He, and J.A. Stankovic. Differentiated surveillance for sensor networks. In *SenSys 2003: Proceedings of the 1st international conference on Embedded networked sensor systems*, pages 51–62, Los Angeles, California, USA, November 5-7 2003. ACM.

[17] K. Chakrabarty, S.S. Iyengar, H. Qi, and E. Cho. Grid coverage for surveillance and target location in distributed sensor networks. *IEEE Transactions on Computers*, 51(12):1448–1453, December 2002.

[18] T. He, S. Krishnamurthy, L. Luo, T. Yan, L. Gu, R. Stoleru, G. Zhou, Q. Cao, P. Vicaire, J.A. Stankovic, T.F. Abdelzaher, J. Hui, and B. Krogh. Vigilnet: An

integrated sensor network system for energy-efficient surveillance. *ACM Transactions on Sensor Networks*, 2(1):1–38, February 2006.

[19] D.K. Nilsson, T. Roosta, U. Lindqvist, and A. Valdes. Key management and secure software updates in wireless process control environments. In *WiSec 2008: Proceedings of the first ACM conference on Wireless network security*, pages 100–108, Alexandria, VA, USA, March 31 - April 2 2008. ACM.

[20] J. Emil, M. Aleksandar, O. Chris, and P. deGroen. A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation. *Journal of NeuroEngineering and Rehabilitation*, 2(1):6, 2005.

[21] B. Gyselinckx, H.C. Van, J. Ryckaert, R.F. Yazicioglu, P. Fiorini, and V. Leonov. Human++: autonomous wireless sensors for body area networks. In *CICC 2005: Proceedings of the IEEE Custom Integrated Circuits Conference 2005*, pages 13–19, San Jose, California, USA, September 18-21 2005.

[22] M. Klemm and G. Troester. Textile uwb antennas for wireless body area networks. *IEEE Transactions on Antennas and Propagation*, 54(11):3192–3197, November 2006.

[23] C. Otto, A. Milenkovic, C. Sanders, and E. Jovanov. System architecture of a wireless body area sensor network for ubiquitous health monitoring. *Journal of Mobile Multimedia*, 1(4):307–326, January 2006.

[24] T. Zasowski, F. Althaus, M. Stager, A. Wittneben, and G. Troster. Uwb for noninvasive wireless body area networks: channel measurements and results. In *Proceedings of the 2003 IEEE Conference on Ultra Wideband Systems and Technologies*, pages 285–289, Reston, Virginia, USA, November 16-19 2003.

[25] N.F. Timmons and W.G. Scanlon. Analysis of the performance of IEEE 802.15.4 for medical sensor body area networking. In *SECON 2004: Proceedings of the First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 16–24, Santa Clara, CA, USA, October 4-7 2004.

[26] S.M.K. Raazi, H. Lee, S. Lee, and Y. Lee. BARI: A distributed key management approach for wireless body area networks. In *CIS 2009: Proceedings of the 2009 International Conference on Computational Intelligence and Security*, pages 324–329, Beijing, China, December 11-14 2009. IEEE Computer Society.

[27] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4(5):42–47, October 1997.

[28] N. Noury, T. Herve, V. Rialle, G. Virone, E. Mercier, G. Morey, A. Moro, and T. Porcheron. Monitoring behavior in home using a smart fall sensor and position sensors. In *Proceedings of the 1st Annual International IEEE-EMBS Special Topic Conference on Microtechnologies in Medicine and Biology*, pages 607–610, Lyon, France, October 12-14 2000.

[29] S.S. Intille. Designing a home of the future. *IEEE Pervasive Computing*, 1(2):76–82, April 2002.

[30] D.J. Cook, M. Youngblood, E.O. Heierman, K. Gopalratnam, S. Rao, A. Litvin, and F. Khawaja. Mavhome: An agent-based smart home. In *PERCOM 2003: Proceedings of the First IEEE International Conference on Pervasive Computing and Communications*, page 521, Dallas-Fort Worth, TX, USA, March 23-26 2003. IEEE Computer Society.

[31] T.S. Barger, D.E. Brown, and M. Alwan. Health-status monitoring through analysis of behavioral patterns. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 35(1):22–27, January 2005.

[32] B. Clarkson, A. Pentland, and K. Mase. Recognizing user context via wearable sensors. In *ISWC 2000: Proceedings of the 4th IEEE International Symposium on Wearable Computers*, page 69, Atlanta, Georgia, USA, October 18-21 2000. IEEE Computer Society.

[33] S.M.K. Raazi, S. Lee, and Y. Lee. TIMAR: an efficient key management scheme for ubiquitous health care environments. In *Mobimedia '09: Proceedings of the 5th International ICST Mobile Multimedia Communications Conference*, pages 1–7, London, United Kingdom, September 7-9 2009. ICST.

[34] J. Szczepanski, E. Wajnryb, J.M. Amig, M.V. Sanchez-Vives, and M. Slater. Biometric random number generators. *Computers & Security*, 23(1):77 – 84, February 2004.

[35] C.C.Y. Poon, Y.T. Zhang, and S.D. Bao. A novel biometrics method to secure wireless body area sensor networks for telemedicine and m-health. *IEEE Communication Magazine*, 44(4):73–81, May 2006.

[36] S. Cherukuri, K.K. Venkatasubramanian, and E.K.S. Gupta. BioSec: A biometric based approach for securing communication. In *ICPPW 2003: Proceedings of the 32nd International Conference on Parallel Processing Workshops*, page 432, Kaohsiung, Taiwan, October 6-9 2003. IEEE Computer Society.

[37] K.V. Krishna and S.K.S Gupta. Security for pervasive health monitoring sensor applications. In *ICISIP 2006: Proceedings of the 4th International Conference*

on Intelligent Sensing and Information Processing, pages 197–202, Bangalore, India, October 15-December 18 2006. IEEE Computer Society.

[38] F.M. Bui and D. Hatzinakos. Biometric methods for secure communications in body sensor networks: Resource-efficient key management and signal-level data scrambling. *EURASIP Journal on Advances in Signal Processing*, vol. 2008:1–16, 2008.

[39] T. Falck, H. Baldus, J. Espina, and K. Klabunde. Plug 'n play simplicity for wireless medical body sensors. *Mobile Networks and Applications*, 12(2-3):143–153, March 2007.

[40] M.A. Hamid, M. Rahman, Y.J. Yoon, and C.S. Hong. Developing a group-based security scheme for wireless sensor networks. In *GLOBECOMM 2007: Proceedings of the IEEE Global Telecommunications Conference 2007*, pages 1354 –1359, Washington, DC, USA, November 26-30 2007.

[41] M.A. Hamid and C.S. Hong. Energy conserving security mechanisms for wireless sensor networks. *Annals of Telecommunications*, 64(11-12):723–734, December 2009.

[42] W. Diffie and M.E. Hellman. New directions in cryptography. *IEEE Transactions on Information Theory*, IT-22(6):644–654, November 1976.

[43] R.L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *Communications of the ACM*, 21(2):120–126, 1978.

[44] T. Dierks and C. Allen. The tls protocol version 1.0, 1999.

[45] J. Kohl and C. Neuman. The kerberos network authentication service (v5), 1993.

[46] A.K. Pathan and C.S. Hong. A key-predistribution-based weakly connected dominating set for secure clustering in dsn. In *HPCC 2006: Proceedings of the 2006 International Conference on High Performance Computing and Communications*, volume 4208 of *Lecture Notes in Computer Science*, pages 270–279, Munich, Germany, September 13-15 2006. Springer Berlin / Heidelberg.

[47] S. Zhu, S. Setia, and S. Jajodia. LEAP+: Efficient security mechanisms for large-scale distributed sensor networks. *ACM Transactions on Sensor Networks*, 2(4):500–528, November 2006.

[48] K.K. Venkatasubramanian, A. Banerjee, and S. Gupta. Plethysmogram-based secure inter-sensor communication in body area networks. In *MILCOM 2008: Proceedings of 2008 IEEE Military Communications Conference*, pages 1–7, San Diego, CA, USA, November 17-19 2008.

[49] K. Ghumman. Location-aware combinatorial key management scheme for clustered sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 17(8):865–882, August 2006. Senior Member-Younis, M.F. and Senior Member-Eltoweissy, M.

[50] M. Eltoweissy, M. Moharrum, and R. Mukkamala. Dynamic key management in sensor networks. *IEEE Communications Magazine*, 44(4):122–130, April 2006.

[51] S.M.K. Raazi, H. Lee, S. Lee, and Y. Lee. MUQAMI+: a scalable and locally distributed key management scheme for clustered sensor networks. *Annals of Telecommunications*, 65(1-2):101–116, February 2010.

[52] C. Karlof, N. Sastry, and D. Wagner. TinySec: a link layer security architecture for wireless sensor networks. In *SenSys 2004: Proceedings of the 2nd international conference on Embedded networked sensor systems*, pages 162–175, Baltimore, MD, USA, November 3-4 2004. ACM.

[53] Y. Xiao, V.K. Rayi, B. Sun, X. Du, F. Hu, and M. Galloway. A survey of key management schemes in wireless sensor networks. *Computer Communications*, 30(11-12):2314 – 2341, September 2007.

[54] W. Xu, K. Ma, W. Trappe, and Y. Zhang. Jamming sensor networks: attack and defense strategies. *IEEE Network*, 20(3):41–47, June 2006.

[55] S.O. Amin, M.S. Siddiqui, and C.S. Hong. Detecting jamming attacks in ubiquitous sensor networks. In *SAS 2008: Proceedings of the IEEE Sensors Applications Symposium 2008*, pages 40–45, Atlanta, GA, USA, February 12-14 2008.

[56] S.O. Amin, M.S. Siddiqui, C.S. Hong, and S. Lee. Rides: Robust intrusion detection system for ip-based ubiquitous sensor networks. *MDPI Sensors*, 9(5):3447–3468, 2009.

[57] S.O. Amin, M.S. Siddiqui, C.S. Hong, and S. Lee. Implementing signature based ids in ip-based sensor networks with the help of signature-codes. *IEICE Transactions on Communications*, E93-B(2):389–391, February 2010.

[58] G. Gupta and M. Younis. Load-balanced clustering of wireless sensor networks. In *ICC 2003: Proceedings of the 2003 IEEE International Conference on Communications*, pages 1848–1852, Anchorage, Alaska, USA, May 11-15 2003.

[59] O. Younis and S. Fahmy. Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. *IEEE Transactions on Mobile Computing*, 3(4):366–379, December 2004.

[60] K. Langendoen and N. Reijers. Distributed localization in wireless sensor networks: A quantitative comparison. *Computer Networks: The International Journal of Computer and Telecommunications Networking*, 43(4):499–518, November 2003.

[61] Youssef A., A. Agrawala, and Younis M. Accurate anchor-free localization in wireless sensor networks. In *WSNIA 2005: Proceedings of the First IEEE Workshop on Information Assurance in Wireless Sensor Networks*, pages 465–470, Phoenix, Arizona, USA, April 7-9 2005.

[62] I. Yoo and M. Song. Biomedical ontologies and text mining for biomedicine and healthcare: A survey. *Journal of Computing Science and Engineering*, 2(2):109–136, June 2008.

[63] A. Celentano, S. Faralli, and F. Pittarello. The situation lens: A metaphor for personal task management on mobile devices. *Journal of Computing Science and Engineering*, 3(4):238–259, December 2009.

[64] E.V. Vasa. Communication timing attack on the public key management system kerberos for distributed authentication through the rsa encryption. Dissertation, The University of Texas at El Paso, United States – Texas, December 2003.

[65] T. Zia and A. Zomaya. Security issues in wireless sensor networks. In *ICSNC 2006: Proceedings of the International Conference on Systems and Networks*

Communications, pages 40–40, Tahiti, French Polynesia, October 29 - November 3 2006. IEEE Computer Society.

[66] C. Karlof, Y. Li, and J. Polastre. Arrive: An architecture for robust routing in volatile environments. Technical Report CSD-03-1233, University of California at Berkeley, 2003.

[67] Y. Wang, G. Attebury, and B. Ramamurthy. A survey of security issues in wireless sensor networks. *IEEE Communications Surveys and Tutorials*, 8(2):2–23, June 2006.

[68] T. Roosta, S. Shieh, and S. Sastry. taxonomy of security attacks in sensor networks and countermeasures. In *SIRI 2006: Proceedings of The First IEEE International Conference on System Integration and Reliability Improvements*, pages 13–15, Hanoi, Vietnam, December 13-15 2006.

[69] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *MOBICOMM 2000: Proceedings of the 6th Annual International Conference on Mobile Computing and Networking*, pages 56–67, Boston, Massachusetts, USA, August 6-11 2000. ACM.

[70] Madden S., M.J. Franklin, Hellerstein J.M., and H. Wei. Tag: a tiny AGgregation service for ad-hoc sensor networks. *ACM SIGOPS Operating Systems Review*, 36(SI):131–146, December 2002.

[71] S. Madden, R. Szewczyk, M.J. Franklin, and D. Culler. Supporting aggregate queries over ad-hoc wireless sensor networks. In *WMCSA 2002: Proceedings of*

the 4th IEEE Workshop on Mobile Computing Systems and Applications, page 49, Callicoon, New York, USA, June 19-20 2002. IEEE Computer Society.

[72] N. Sultana, M.C. Ki, and E. Huh. Application driven cluster based group key management with identifier in mobile wireless sensor network. In *FGCN 2007: Proceedings of the 2007 International Conference on Future Generation Communication and Networking*, pages 362 –367, Jeju Island, Korea, December 6-8 2007. IEEE Computer Society.

[73] E. Huh and N. Sultana. Application Driven Cluster Based Group Key Management with Identifier in Mobile Wireless Sensor Networks. *KSII Transactions on Internet and Information Systems(TIIS)*, 1(1):5–18, December 2007.

[74] H. Chan, A. Perrig, and D. Song. Random key predistribution schemes for sensor networks. In *SP 2003: Proceedings of the 2003 IEEE Symposium on Security and Privacy*, pages 197–213, Oakland, California, USA, May 11-13 2003. IEEE Computer Society.

[75] L. Eschenauer and V.D. Gligor. A key-management scheme for distributed sensor networks. In *CCS 2002: Proceedings of the 9th ACM conference on Computer and communications security*, pages 41–47, Washington, DC, USA, November 18-22 2002. ACM.

[76] D. Liu and P. Ning. Establishing pairwise keys in distributed sensor networks. In *CCS 2003: Proceedings of the 10th ACM Conference on Computer and Communications Security*, pages 52–61, Washington D.C., USA, October 27-30 2003. ACM.

[77] M.M. Haque, A.K. Pathan, C.S. Hong, and E. Huh. An asymmetric key-based security architecture for wireless sensor networks. *KSII Transactions on Internet and Information Systems(TIIS)*, 2(5):265–279, October 2008.

[78] D.J. Malan, M. Welsh, and M.D. Smith. A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography. In *SECON 2004: First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks*, pages 71–80, Santa Clara, CA, USA, October 4-7 2004.

[79] N. Gura, A. Patel, W. Arvinderpal, H. Eberle, and S.C. Shantz. Comparing elliptic curve cryptography and rsa on 8-bit cpus. In *CHES 2004: Proceedings of the 6th International Workshop on Cryptographic Hardware and Embedded Systems*, pages 119–132, Cambridge (Boston), USA, August 11-13 2004. Springer-Verlag.

[80] A.S. Wander, N. Gura, H. Eberle, V. Gupta, and S.C. Shantz. Energy analysis of public-key cryptography for wireless sensor networks. In *PERCOM 2005: Proceedings of the 3rd IEEE International Conference on Pervasive Computing and Communications*, pages 324–328, Kauai, Hawaii, USA, March 8-12 2005. IEEE Computer Society.

[81] Q. Huang, J. Cukier, H. Kobayashi, B. Liu, and J. Zhang. Fast authenticated key establishment protocols for self-organizing sensor networks. In *WSNA 2003: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 141–150, San Diego, CA, USA, September 19 2003. ACM.

[82] P. Kotzanikolaou, E. Magkos, D. Vergados, and M. Stefanidakis. Secure and practical key establishment for distributed sensor networks. 2(6):595–610, February 2009.

[83] R.A. Shaikh, S. Lee, M.A.U. Khan, and Y. Song. LSec: Lightweight security protocol for distributed wireless sensor networks. In *PWC 2006: In Proceedings of the 11th IFIP International Conference on Personal Wireless Communications*, volume 4217 of *Lecture Notes in Computer Science*, pages 367–377, Albacete, Spain, September 20-22 2006. Springer.

[84] E. Eltoweissy, M.H. Heydari, L. Morales, and I.H. Sudborough. Combinatorial optimization of group key management. *Journal of Network and Systems Management*, 12(1):33–50, March 2004.

[85] D. Eastlake and P. Jones. Us secure hash algorithm 1 (sha1), 2001.

[86] R. Rivest. The md5 message-digest algorithm, 1992.

[87] S. Zhu, S. Setia, and S. Jajodia. LEAP: efficient security mechanisms for large-scale distributed sensor networks. In *CCS 2003: Proceedings of the 10th ACM conference on Computer and communications security*, pages 62–72, Washington D.C., USA, October 27-30 2003. ACM.

[88] A. Banerjee, K. Venkatasubramanian, and S.K.S. Gupta. Challenges of implementing cyber-physical security solutions in body area networks. In *BodyNets 2009: Proceedings of the Fourth International Conference on Body Area Networks*, April 1-3 2009.

[89] A. Juels and M. Sudan. A fuzzy vault scheme. *Designs, Codes and Cryptography*, 38(2):237–257, February 2006.

[90] G. Li, J. He, and Y. Fu. A hexagon-based key predistribution scheme in sensor networks. In *ICPPW 2006: Proceedings of the 2006 International Conference on Parallel Processing Workshops*, pages 175–180, Columbus, Ohio, USA, August 14-18 2006. IEEE Computer Society.

[91] W. Du, J. Deng, Y.S. Han, and P.K. Varshney. A pairwise key pre-distribution scheme for wireless sensor networks. In *CCS 2003: Proceedings of the 10th ACM conference on Computer and communications security*, pages 42–51, Washington D.C., USA, October 27-30 2003. ACM.

[92] T.T. Dai, A.K. Pathan, and C.S. Hong. A resource-optimal key pre-distribution scheme with enhanced security for wireless sensor networks. In *APNOMS 2006: Proceedings of the 9th Asia-Pacific Network Operations and Management Symposium*, volume 4238 of *Lecture Notes in Computer Science*, pages 546–549, Busan, Korea, September 27-29 2006. Springer Berlin/Heidelberg.

[93] T.T. Dai, C.T. Hieu, and C.S. Hong. An efficient id-based bilinear key predistribution scheme for distributed sensor networks. In *HPCC 2006: Proceedings of the 2006 International Conference on High Performance Computing and Communications*, volume 4208 of *Lecture Notes in Computer Science*, pages 260–269, Munich, Germany, September 13-15 2006. Springer Berlin / Heidelberg.

[94] T.T. Dai, C.T. Hieu, M.M. Rahman, and C.S. Hong. A robust pairwise key predistribution scheme resistant to common attacks for wireless sensor networks. In *WISA 2006: Proceedings of the 7th International Workshop on Information Security Applications*, volume 4298 of *Lecture Notes in Computer Science*, pages 121–135, Jeju Island, Korea, August 28-30 2006. Springer.

[95] N.T.T. Huyen and E. Huh. An efficient signal range based key pre-distribution scheme ensuring the high connectivity in wireless sensor network. In *ICUIMC 2008: Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pages 441–447, Suwon, Korea, January 31 - February 1 2008. ACM.

[96] H.T.T. Nguyen, M. Guizani, Minho Jo, and Eui-Nam Huh. An efficient signal-range-based probabilistic key predistribution scheme in a wireless sensor network. *IEEE Transactions on Vehicular Technology*, 58(5):2482–2497, june 2009.

[97] T.T. Dai and C.S. Hong. Efficient id-based threshold random key pre-distribution scheme for wireless sensor networks. *IEICE Transactions on Communications*, E91-B(8):2602–2609, August 2008.

[98] A.K. Pathan, T.T. Dai, and C.S. Hong. An efficient lu decomposition-based key pre-distribution scheme for ensuring security in wireless sensor networks. In *CIT 2006: Proceedings of the Sixth IEEE International Conference on Computer and Information Technology*, pages 227–232, September 20-22 2006.

[99] A.K. Pathan, T.T. Dai, and C.S. Hong. A key management scheme with encoding and improved security for wireless sensor networks. In *ICDCIT 2006: Proceedings of the The 3rd International Conference on Distributed Computing and Internet Technology*, pages 102–115, Bhubaneswar, India, December 20-23 2006. Springer.

[100] S.A. Çamtepe and B. Yener. Combinatorial design of key distribution mechanisms for wireless sensor networks. *IEEE/ACM Transactions on Networking*, 15(2):346–358, 2007.

[101] G. Dini and I.M. Savino. An efficient key revocation protocol for wireless sensor networks. In *WOWMOM 2006: Proceedings of the 2006 International Symposium on on World of Wireless, Mobile and Multimedia Networks*, pages 450–452, Niagara-Falls, Buffalo-NY, USA, June 26-29 2006. IEEE Computer Society.

[102] K. Paek, J. Kim, C. Hwang, and U. Song. An energy-efficient key management protocol for large-scale wireless sensor networks. In *MUE 2007: Proceedings of the 2007 International Conference on Multimedia and Ubiquitous Engineering*, pages 201–206, Seoul, Korea, April 26-28 2007. IEEE Computer Society.

[103] L. Lamport. Password authentication with insecure communication. *Communicartions of the ACM*, 24(11):770–772, November 1981.

[104] A.J. Menezes, S.A. Vanstone, and P.C.V. Oorschot. *Handbook of Applied Cryptography*. CRC Press, Inc., Boca Raton, FL, USA, 1996.

[105] G. Xing, C. Lu, Y. Zhang, Q. Huang, and R. Pless. Minimum power configuration in wireless sensor networks. In *MobiHoc 2005: Proceedings of the 6th ACM international symposium on Mobile ad hoc networking and computing*, pages 390–401, Urbana-Champaign, IL, USA, May 25-28 2005. ACM.

[106] R. Venugopalan, P. Ganesan, P. Peddabachagari, A. Dean, F. Mueller, and M. Sichitiu. Encryption overhead in embedded systems and sensor network nodes: modeling and analysis. In *CASES 2003: Proceedings of the 2003 international conference on Compilers, Architecture and Synthesis for Embedded Systems*, pages 188–197, San Jose, California, USA, October 30 - November 1 2003. ACM.

[107] D. Seetharam and S. Rhee. An efficient pseudo random number generator for low-power sensor networks. In *LCN 2004: Proceedings of the 29th Annual IEEE International Conference on Local Computer Networks*, pages 560–562, Tampa, FL, USA, November 16-18 2004. IEEE Computer Society.

[108] H. Qiang, C. Johnas, K. Hisashi, L. Bede, and Z. Jinyun. Fast authenticated key establishment protocols for self-organizing sensor networks. In *WSNA 2003: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, pages 141–150, San Diego, CA, USA, September 19 2003. ACM.

[109] S.M.K. Raazi, A.M. Khan, F.I. Khan, S.Y. Lee, and Y.J. Song. MUQAMI: A locally distributed key management scheme for clustered sensor networks. In *IFIPTM 2007: Joint iTrust and PST Conferences on Privacy, Trust Management and Security*, volume Volume 238/2007 of *IFIP International Federation for Information Processing*, pages 333–348, Moncton, New Brunswick, Canada, July 30 - August 2 2007. Springer Boston.

[110] R. Moharram, R. Mukkamala, and M. Eltoweissy. TKGS: Verifiable threshold-based key generation scheme in open wireless ad hoc networks. In *ICCCN 2004: Proceedings of the 13th IEEE International Conference on Computer Communication and Networking*, pages 31–36, Rosemont, IL, USA, October 11-13 2004. IEEE.

[111] R.A. Shaikh, H. Jameel, B.J. d'Auriol, H. Lee, S. Lee, and Y. Song. Group-based trust management scheme for clustered wireless sensor networks. *IEEE Transactions on Parallel and Distributed Systems*, 20(11):1698–1712, 2009.

[112] L.Y. Wei, V.H. Lodewijk, D. Jeroen, H. Pieter, and H. Paul. Energy-efficient link-layer jamming attacks against wireless sensor network mac protocols. In *SASN 2005: Proceedings of the The 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks*, pages 76–88, Alexandria, VA, USA, November 7 2005. ACM.

[113] H. Sun, S. Hsu, and C. Chen. Mobile jamming attack and its countermeasure in wireless sensor networks. In *AINAW 2007: Proceedings of the 21st International Conference on Advanced Information Networking and Applications Workshops*, volume 1, pages 457–462, Niagara Falls, Ontario, Canada, May 21-23 2007. IEEE Computer Society.

[114] R.A. Shaikh, H. Jameel, B.J. d'Auriol, H. Lee, S. Lee, and Y. Song. Achieving network level privacy in wireless sensor networks. *MDPI Sensors*, 10(3):1447–1472, 2010.

Publications

Book Chapters

1. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Zeeshan Pervez and Sungyoung Lee, "Key Management Schemes of Wireless Sensor Networks: A Survey", *Security of Self-Organizing Networks: MANET, WSN, WMN, VANET*, Auerbach Publications, CRC Press, Taylor & Francis Group, USA, 2009.

Journals

1. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Heejo Lee, Sungyoung Lee and Young-Koo Lee, "BARI+: A Biometric Based Distributed Key Management Approach for Wireless Body Area Networks", *Sensors Journal (ISSN 1424-8220)*, Volume 10, Number 4, MDPI, April 2010, pp. 3911-3933.

2. **Syed Muhammad Khaliq-ur-Rahman Raazi**, and Sungyoung Lee, "A Survey on Key Management Strategies for Different Applications of Wireless Sensor Networks", *Journal of Computing Science and Engineering (ISSN: 1976-4677)*, Volume 4, Number 1, Korean Institute of Information Scientists and Engineers(KIISE), March 2010, pp. 23-51.

3. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Heejo Lee, Sungyoung Lee and Young-Koo Lee, "MUQAMI+: A Scalable and Locally Distributed Key Management Scheme for Clustered Sensor Networks", *Annals of Telecommunications (ISSN: 0003-4347)*, Volume 65, Number 1-2, Springer, Paris, February 2010, pp. 101-116.

4. Faraz Idris Khan, Hassan Jameel, **Syed Muhammad Khaliq-ur-Rahman Raazi**, Adil Mehmood Khan and Eui Nam Huh, "An Efficient Re-keying Scheme for Cluster Based Wireless Sensor Networks", *Lecture Notes in Computer Science (ISSN: 0302-9743)*, Volume 4706, Springer Berlin / Heidelberg, August 2007, pp. 1028-1037.

Conferences

1. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Heejo Lee, Sungyoung Lee and Young-Koo Lee, "BARI: A Distributed Key Management Approach for Wireless Body Area Networks", in proc. *of the 2009 International Conference on Computational Intelligence and Security (CIS 2009)*, ISBN: 978-0-7695-3931-7, vol. 2, Beijing, China, December 11-14, 2009, pp. 324-329.

2. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Sungyoung Lee and Young-Koo Lee, "TIMAR: An Efficient Key Management Scheme for Ubiquitous Health Care Environments", in proc. *of the 5th International Mobile Multimedia Communications Conference (Mobimedia 2009)*, London, United Kingdom, September 7-9, 2009.

3. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Sungyoung Lee and Young-Koo Lee, "A Novel Architecture for Efficient Key Management in Humanware Applications", in proc. *of the 5th International Joint Conference on INC, IMS and IDC*

(NCM 2009), ISBN: 978-0-7695-3769-6, Seoul, Korea, August 25-27, 2009, pp. 1918-1922.

4. L. Hung, Riaz A. Shaikh, Hassan J., **S.M.K. Raazi**, Y. Weiwei, N. Canh, P. Truc, S. Lee, H. Lee, Y. Son, and M. Fernandes, "Activity Oriented Access Control for Ubiquitous Environments", in proc. *of the 6th Annual IEEE Consumer Communications & Networking Conference (CCNC 2009)*, Las Vegas Jan, 2009, pp. 1-5.

5. L. Hung, Hassan J., Riaz A. Shaikh, **S.M.K. Raazi**, Y. Weiwei, N. Canh, P. Truc, S. Lee, H. Lee, Y. Son, and M. Fernandes, "Activity-based Security Scheme for Ubiquitous Environments", in proc. *of 27th IEEE International Performance Computing and Communications Conference* , USA, Dec 2008, pp. 475-481.

6. Hassan Jameel, Riaz A. Shaikh, Le Xuan Hung, Yuan WeiWei, **Syed Muhammad Khaliq-ur-rehman Raazi**, Ngo Trong Canh, Sungyoung Lee, Heejo Lee, Yuseung Son and Miguel Fernandes, "Image-Feature based Human Identification Protocols on Limited Display Devices", In proc. of *the 9th International Workshop on Information Security Applications (WISA 2008)*, Jeju, korea, Sep 2008, pp. 211-224.

7. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Adil Mehmood Khan, Faraz Idris Khan, Sung Young Lee, Young Jae Song and Young-Koo Lee, "MUQAMI: A Locally Distributed Key Management Scheme for Clustered Sensor Networks", in proc. *of the Joint iTrust and PST conferences on privacy, trust management and security (IFIPTM 2007)*, ISBN: 978-0-387-73654-9, vol. 238, Moncton, New Brunswick, Canada, July 30 - August 02 2007, pp. 333-348.

8. Brian J. d'Auriol, Nguyen Thi Thanh Tuyen, Vo Quoc Hung, Duc Thang, Riaz Ahmed Shaikh, Hassan Jameel, Le Xuan Hung, **S.M.K.R. Raazi**, Dao Phuong

Thuy, Ngo Trong Canh, Adil Mehmood Khan, Sunghyun Kim, Shu Lei, Sakib Pathan, Tran Van Phuong, Sungyoung Lee and Young-Koo Lee, "Embedded Processor Security", in proc. *of the 2007 International Conference on Security and Management (SAM'07)*, Monte Carlo Resort, Las Vegas, Nevada, USA, June 25-28, 2007.

9. **Syed Muhammad Khaliq-ur-Rahman Raazi**, Sung Young Lee, Young Jae Song and Young-Koo Lee, "A novel architecture for localized key management in wireless sensor networks", in proc. *of the 27th KIPS Spring conference*, Kyungwon University, Korea, May 11-12, 2007, pp. 1091-1092.

List of Abbreviations and Notations

WSN	Wireless Sensor Network
WBAN	Wireless Body Area Network
CN	Command Node or the Base Station
MS	Medical Server
PS	Personal Server
CH	Cluster Head Node
KG	Key Generating Node
SN	Sensor Node