January 2013

Prevention and Detection of Intrusions in Wireless Sensor Networks

Ismail Butun
University of South Florida, ibutun@mail.usf.edu

Prevention and Detection of Intrusions in Wireless Sensor Networks

by

Ismail Butun

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
Department of Electrical Engineering
College of Engineering
University of South Florida

Major Professor: Ravi Sankar, Ph.D.
Salvatore D. Morgera, Ph.D.
Sanjukta Bhanja, Ph.D.
Jarred Ligatti, Ph.D.
Xiang-Dong Hou, Ph.D.

Date of Approval:
March 26, 2013

Keywords: network security, access control, user authentication, cryptography, clustering

DEDICATION

This dissertation is dedicated to my deceased father Orhan Bütün.

May God let him rest in peace ...

ACKNOWLEDGMENTS

I am truly indebted to my supervisor, Dr. Ravi Sankar, for his guidance, support, and constant encouragement during my Ph.D. study. I am grateful to Dr. Salvatore D. Morgera, Dr. Sanjukta Bhanja, Dr. Jarred Ligatti and Dr. Xiang-dong Hou, for serving in my committee; and for their precious time and invaluable suggestions. I want to thank Dr. Autar Kaw for chairing my defense. I would also like to acknowledge Dr. Paris Wiley, Dr. Kenneth A. Buckle, and staff from the Department of Electrical Engineering at USF, who have always been very helpful, supportive and kind.

I am also grateful to my dear friends and colleagues at the Interdisciplinary Communication Networking and Signal Processing (*i*CONS) group. I am particularly grateful to my friends; Özgür Yürür, Ali Görçin, Dr. Murad Khalid, Dr. Kun Li, Dr. Xuan Hung Le, Dr. In-ho Ra, Vanitha Narayan Raju, Dr. Mustafa Emin Şahin, Dr. Hisham Mahmood, Dr. Tevfik Yücek, Dr. Hasari Çelebi, and Dr. Sabih Güzelgöz for their encouragement, support, collaboration and enlightening discussions.

Words are not enough to express deepest gratitude to my parents Orhan-Emine Bütün, grandparents Fahri-Aysel Öztürk, elder brother Ömer Faruk Bütün, cousins Saliha-Esma Eroğlu, uncle-auntie Selahaddin-Gülsen Eroğlu; for their relentless support, encouragement and supplications throughout my life.

I would not find enough courage to finish Ph.D. without the moral support of my friends, therefore I would like to express my special thanks to them: Dr. Mehmet Demirer, Mustafa Aydın, Dr. Celal Çeken, Dr. Turgay Temel, Furkan Akın, Erhan Agırman, Murat Altıparmak, Şafak Çırağı, Orhan Ayran, A. Emre Tiryaki, Ahmet Demirçin, Kubilay Özkardeşler, Mutlu Uslu, Ferhat Uçan, Koray İnçki, M. Serdar Soran, Bilge E. Soran, Oğuz Tezcan, Murat Gürdal, Ergün Ünsal, Bolat Dinç, Alper Kültür, Kağan Dağdeviren, Mahmut Gülnaz, İlteriş Mirzaoğlu, Aytekin Kale, and Üveys Danış.

In the end, the ultimate reality is that this would not be possible without the Will of God. I pray to God to pave our way to ultimate knowledge and help us use it for the real benefit of mankind.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

Wireless Sensor Networks (WSNs) continue to grow as one of the most exciting and challenging research areas of engineering. They are characterized by severely constrained computational and energy resources and also restricted by the ad-hoc network operational environment. They pose unique challenges, due to limited power supplies, low transmission bandwidth, small memory sizes and limited energy. Therefore, security techniques used in traditional networks cannot be directly adopted. So, new ideas and approaches are needed, in order to increase the overall security of the network. Security applications in such resource constrained WSNs with minimum overhead provides significant challenges, and is the main focus of this dissertation.

There is no "one size fits all" solution in defending WSNs against intrusions and attacks. Therefore, intrusions and attacks against WSNs should be carefully examined to reveal specific vulnerabilities associated with them, before beginning the design of any kind of intrusion prevention and detection systems. By following this rationale, the dissertation starts with providing information regarding the WSNs, types of attacks towards WSNs, and the methods on how to prevent and detect them. Then, in order to secure WSNs, a security provisioning plan is provided.

In general, the following processes may be involved in securing WSNs: Intrusion Prevention, Intrusion Detection, and Intrusion Mitigation. This dissertation presents solutions (algorithms and schemes) to the first two lines of defenses of the security provisioning plan, namely, Intrusion Prevention and Intrusion Detection.

As a first line of defense in securing WSNs, this dissertation presents our proposed algorithm ("Two-Level User Authentication" scheme) as an Intrusion Prevention System (IPS) for WSNs. The algorithm uses two-level authentication between a sensor node and a user. It is designed for heterogeneous WSNs, meaning that the network consists of two components: regular nodes and more powerful cluster heads. The proposed scheme is evaluated both analytically and also in a simulation environment, by comparing it to the current state-of-the-art schemes in the literature.

A comprehensive and systematic survey of the state-of-the-art in Intrusion Detection Systems (IDSs) that are proposed for Mobile Ad-Hoc Networks (MANETs) and WSNs is presented. Firstly, detailed information about IDSs is provided. This is followed by the analysis and comparison of each scheme along with their advantages and disadvantages from the perspective of security. Finally, guidelines on IDSs that are potentially applicable to WSNs are provided. Overall, this work would be very helpful to the researchers in developing their own IDSs for their WSNs.

Clustering (of the nodes) is very important for WSNs not only in data aggregation, but also in increasing the overall performance of the network, especially in terms of total life-time. Besides, with the help of clustering, complex intrusion prevention and detection algorithms can be implemented. Therefore, background on the clustering algorithms is provided and then a clustering algorithm for WSNs is proposed, that is both power and connectivity aware. The proposed algorithm provides higher energy efficiency and increases the life-time of the network. In evaluating the proposed clustering algorithm (in a simulation environment by comparing its' performance to the previously proposed algorithm, namely Kachirski *et al.*'s algorithm), it is demonstrated that the proposed algorithm improves energy efficiency in WSNs.

Finally, an IDS framework based on multi-level clustering for hierarchical WSNs is proposed. It is based upon (the nodes use our proposed clustering algorithm while forming their clusters) the clustering algorithm that is proposed in this dissertation. The framework provides two types of intrusion detection approaches, namely "Downwards-IDS (D-IDS)" to detect the abnormal behavior (intrusion) of the subordinate (member) nodes and "Upwards-IDS (U-IDS)" to detect the abnormal behavior of the cluster heads. By using analytical calculations, the optimum parameters for the D-IDS (number of maximum hops) and U-IDS (monitoring group size) of the framework are evaluated and presented.

Overall, this dissertation research contributes to the first two lines of defenses towards the security of WSNs, namely, IPS and IDS. Furthermore, the final contribution of this dissertation is towards the topology formation of the WSNs (especially for the hierarchical WSNs), namely, clustering; which would be very useful in implementation of the IPS and IDS systems that are presented in this dissertation.

CHAPTER 1 :

INTRODUCTION

1.1 Background

Wireless communications is inevitable in today's technology owing to the advantages it brings, such as mobility, portability, freedom from wired infrastructure, etc. Despite the benefits, it introduces opportunities to adversaries for eavesdropping of the data being transmitted, and also makes active intrusions easier (through the wireless medium). In order to prevent unauthorized access to the network, the design of secure communication protocols are needed, which will provide both privacy of the wireless data communications and authenticity of communicating parties.

Wireless Sensor Networks (WSNs) continue to grow as one of the most exciting and challenging research areas of engineering. There are many applications of WSNs which are intended to monitor physical and environmental phenomena such as ocean and wildlife, earthquakes, pollution, wild fires and water quality. WSNs can also be used to gather information regarding human activities such as health care, manufacturing machinery performance, building safety, military surveillance and reconnaissance, highway traffic, etc.

WSNs are characterized by severely constrained computational and energy resources, and an ad hoc operational environment. They possess unique characteristics such as limited power supplies, low transmission bandwidth, small memory size and limited energy; therefore security techniques used in traditional networks cannot be adopted directly. Security applications (e.g. intrusion prevention and intrusion detection) in such resource constrained WSNs with minimum overhead reveals significant challenges, and is the main focus of this dissertation.

As mentioned earlier, WSNs are one of the most promising technologies that have applications ranging from health care to tactical military. Although WSNs have appealing features (e.g. low installation cost, unattended network operation, etc.), due to the lack of a physical line of defense (i.e., there are no gateways or switches to monitor the information flow) and also due to the physical constraints, the security of such networks is a big concern. This is valid for the applications especially

where confidentiality has prime importance. For instance, securing WSNs is critically important in tactical (military) applications where a security gap in the network would cause casualties of the friendly forces in a battlefield.

Stuxnet virus, which was targeted at the Iranian nuclear power plants, has shown that the network security is not a hoax, but a reality [1]. A 500KB worm has shut down a number of facilities (14 industrial sites) and severely damaged targeted power plants resulting in up to 2 years of delay to the uranium-enrichment plan. One of the intriguing facts of the Stuxnet is that it was able to penetrate closed networks (intranet), although the targeted facilities did not have direct access to the internet. This is a very important example from real life that shows us the importance of preventing and also detecting intrusions in any kind of network (closed or open) on time.

In order to operate WSNs in a secure way, if possible, intrusions should be prevented with intrusion prevention techniques. Otherwise, they should be detected on time with intrusion detection techniques, before attackers can harm the network resources (i.e., sensor nodes) and/or information destination (i.e., data sink or base station).

1.2 Research Motivation and Goals

WSNs are characterized by severely constrained computational and energy resources, and also restricted by the ad-hoc network operational environment. Although there are plenty of applications for the WSNs, in most cases they are deployed in hostile environments where physical security does not exist and they are operated in an un-attended way.

Keeping these constraints of the WSNs in mind, it is obvious to conclude that traditional security solutions of wired/wireless networks would not be feasible for WSNs. Any security solution to be devised for WSNs needs to consider these limitations and constraints of WSNs. These facts encouraged us to address security challenges in our dissertation as one of the main concerns.

Intrusion prevention is the first line of defense in any security system. Therefore, any security plan being developed for WSNs should start with an Intrusion Prevention System (IPS) that is suitable (designed by taking into account of all the constraints and challenges of WSNs). Current IPSs available in the literature, SPINS [2], TinySec [3] and LEAP [4], provide only one-hop "node authentication", opting out the end-to-end secure communication (i.e., between the user and the sensor node). This is a very big security drawback, since the integrity of the data being transmitted cannot be guaranteed. Other proposed schemes provide end-to-end secure communication with

either Secret Key Cryptography (SKC) [5–7]; or with Public Key Cryptography (PKC) [8]. SKC based schemes are not scalable for thousands of sensor nodes and users, and need significant memory to store authentication codes. Thus, addition of new nodes and users is troublesome in terms of key distribution. On the contrary, PKC based scheme is not practical for WSNs because of the homogenous network structure it possess, meaning that all the power and processing demanding PKC operations are supposed to be handled on the normal sensor nodes. As a result, authentication operations take minutes and batteries of the sensor nodes deplete faster. Therefore, one of the main goals of this dissertation is to propose an *IPS* that provides a unique solution (by using PKC and SKC in an intelligent way) to prevent intrusions in WSNs by opting out all the design drawbacks mentioned above.

When an intruder manages to pass the first line of defense (namely IPS) in a network, it should be detected by the Intrusion Detection System (IDS); in order to take further action to diminish the damages that could be carried out by the intruder. Hence, IDS should constitute the second line of defense in securing the WSNs. Here, it is important to emphasize that our focus (of IDS) is on the hierarchical WSNs, meaning that sensor nodes are gathered into groups called "Clusters". The current IDSs for hierarchical WSNs available in the literature have drawbacks: In the IDS approaches proposed by [9], [10] and [11], the direction of the alert propagation is from sub-ordinates through CHs, leaving the following question unanswered for the detection part: "What happens if a malicious CH drops the packet that is coming from a subordinate node and is about to alert an upper level CH?". In the IDS approaches proposed by Agah *et al.* [12, 13], only one of the clusters of the network is monitored at a time. This leaves the rest of the network unprotected. In the IDS approach of Su *et al.* [14], both downwards and upwards protection are provided, meaning that CH's monitor subordinate nodes and vice versa, respectively. However, the proposed scheme uses SKC and therefore new nodes cannot be added to the network after the deployment, which makes it impractical. Therefore, another main goal of this dissertation is to propose an *IDS framework for hierarchical WSNs* that provides a unique solution to detect intrusions in WSNs by opting out all the design drawbacks mentioned above.

Hierarchical WSNs usually use "Clustering" for the formation of their architecture. With the help of clustering, complex intrusion prevention and detection algorithms can be implemented. Furthermore, clustering is very important for WSNs not only in data aggregation, but also in increasing the overall performance of the network, especially in terms of total life-time. Clustering algorithms

for WSNs should be devised considering the special needs of WSNs; most importantly they should be very stingy in power consumption. Additionally, since cluster heads are assigned with high power consuming tasks, clustering algorithm should take into account the power levels of each candidate. Finally, elected cluster heads should be somewhere close to the main communication hubs, thus clustering algorithm should also consider the connectivities of the candidates. The proposed clustering algorithms in the literature either do not consider power awareness [15, 16], connectivity awareness [17–21], or both [9, 22, 23]. Therefore, another goal of this dissertation is to design a *Clustering Algorithm* that will be aware of both the power and the connectivity of the candidate nodes.

1.3 Contributions of this Dissertation

In providing "Security Provisioning" for WSNs, our main contributions in the different areas can be summarized as follows:

1. *Intrusion Prevention:* For Intrusion Prevention in WSNs, we have developed a Two Level User Authentication scheme. Our scheme is designed for heterogeneous WSNs, where the network consists of two different types of elements, namely cluster heads and sensor nodes. Our analysis and simulation results show that our scheme is not only more secure and scalable than existing secret key cryptography based schemes [5–7], but also requires less processing power and provides higher energy efficiency than existing public key cryptography based schemes [8].

2. *Intrusion Detection:* For Intrusion Detection in WSNs, we have developed a framework to detect intrusions in WSNs. Our framework is an IDS based on multi-level clustering for hierarchical WSNs. The framework consists of two parts: 1)Downwards-IDS (D-IDS) and 2) Upwards-IDS (U-IDS). D-IDS detects intrusions through subordinate members, whereas U-IDS detects intrusions through cluster heads. By using analytical calculations, the optimum parameters (number of maximum hops for D-IDS and monitoring group size for the U-IDS) of our proposed framework are evaluated.

3. *Clustering:* We have devised a power and connectivity aware clustering algorithm that increases energy efficiency, and therefore increases the overall life-time of the WSNs. According to the simulation results, our proposed algorithm is energy efficient and also provides longer life-time to the network (in the worst-case scenario, up to 85% for a 15-node

network with 1-hop connectivity), compared to the previous algorithm (namely, Kachirski *et al.*'s algorithm [15]).

1.4 Organization of this Dissertation

The organization of this dissertation is as follows:

In Chapter 2[1], a detailed background on WSNs is provided. Chapter 2 starts with the description of wireless ad-hoc networks (mobile ad-hoc networks - MANETs), and then provides the characteristics of WSNs and finally emphasizes the distinctions between WSNs and MANETs. Afterwards, the application areas of the WSNs are mentioned. Finally, the security issues for WSNs, such as the attacks against WSN security are discussed. Without knowledge of attacks, neither can security measures be devised to protect WSNs, nor can models be developed to detect intrusions towards the WSNs. Therefore, more information on attacks and security measures to counter those attacks are provided. One of the main counter measures against attacks is *Cryptography*. Some background of cryptography and its application to WSNs are provided. Following that, some open problems in WSN security are mentioned. Finally, the authors' point of view regarding the provisioning of the security towards WSNs is presented.

In Chapter 3[2], a detailed description of the proposed Two Level User Authentication scheme for heterogeneous WSNs is provided. It is an IPS that was devised to prevent intrusions against WSNs. The proposed user authentication scheme is secure and scalable. In addition, it employs both public and secret key cryptography schemes, by taking advantage of the strengths of both schemes. In order to evaluate security and performance analysis of the proposed scheme, it is compared to the current state-of-the-art schemes in the literature, both analytically and with simulations.

In Chapter 4[3], a thorough literature survey of the state-of-the-art IDSs that are proposed for WSNs is provided. Firstly, detailed information about IDSs is provided. Secondly, a brief survey of IDSs proposed for Mobile Ad-Hoc Networks (MANETs) is presented and applicability of those systems to WSNs is discussed. Thirdly, IDSs proposed for WSNs are presented. This is followed by the analysis and comparison of each scheme along with their advantages and disadvantages. Finally, guidelines on IDSs that are potentially applicable to WSNs are provided. The chapter concludes by highlighting open research issues in the field.

[1]The content of this chapter is published in parts in [24, 25].
[2]The content of this chapter is published in parts in [26–28].
[3]The content of this chapter is published in parts in [29].

In Chapter 5[4], clustering algorithms that are proposed for WSNs are investigated. Then, a clustering algorithm for WSNs, that is both power and connectivity aware is proposed. The proposed algorithm provides higher energy efficiency and increases the life-time of the network. The proposed clustering algorithm is evaluated in a simulation environment and its' performance to a previously proposed algorithm (namely Kachirski *et al.*'s algorithm) is compared.

Chapter 6[5] presents our proposed IDS framework for hierarchical WSNs that is based on multi-level clustering. It is based upon the clustering algorithm that is proposed in this dissertation (the nodes use our proposed clustering algorithm that is presented in Chapter 5, while forming their clusters). Our proposed IDS framework provides two types of intrusion detection approaches, namely "Downwards-IDS (D-IDS) Scheme" to detect the abnormal behavior (intrusion) of the subordinate (member) nodes and "Upwards-IDS Scheme" to detect the abnormal behavior of the cluster heads. Furthermore, the effect of cluster size (maximum hops between cluster head and cluster members) on the detection (malicious subordinate nodes) probability of the proposed D-IDS scheme is evaluated. Finally, the effect of total number of monitoring nodes on the detection (malicious cluster head) probability of the proposed U-IDS scheme is evaluated.

In Chapter 7, contributions from Chapters 2-6 are summarized and then recommendations for future work are presented.

[4]The content of this chapter is published in parts in [30].
[5]The content of this chapter is published in parts in [31].

CHAPTER 2 :

SECURITY IN WIRELESS SENSOR NETWORKS

Wireless ad hoc network (or Mobile ad hoc network - MANET[1]) is composed of a network of wireless devices that have no a priori infrastructure support (there is no specifically assigned routers or gateways exists). These devices in this context are called "nodes". Nodes dynamically establish connections when they are in radio range of one another and thus this is called communication "on the fly". Nodes that are out of the range of each other rely on the intermediate nodes to forward their packets. Therefore, each node may act as source, sink or a relay for packets depending on the position [32].

MANETs are multi-hop networks in which all nodes work cooperatively to maintain network connectivity. They are useful in situations where temporary network connectivity is needed such as natural disaster area. Such a network would allow medical personnel to retrieve patient records from hospital databases assuming that the network managing station (base station) of the network has connection to those databases via internet or some other ways. In the same manner insurance company agents can get and submit queries from their databases in order to file claims regarding to the damages of their customer goods.

Recent developments in wireless communications and micro electro mechanical systems (MEMS) technologies facilitated the design of wireless sensor networks (WSNs), in which sensor nodes collect the intelligible data from their surrounding environments and share them in a wireless fashion to send the information towards a meaningful data sink. WSNs are special application of MANETs, in which they have limitations on energy, computational power, memory storage, mobility, etc.

According to scientific predictions, the total number of wireless sensors deployed is expected to reach 60 trillion in years 2012-2022, meaning 10,000 sensors for every person on the world [33]. Therefore, all the problems and challenges concerning WSNs will expose plentiful topics for academia as well as the commercial researchers.

[1]In general, the terms of mobile ad-hoc networks (MANETs) and wireless ad-hoc networks are used interchangeably; therefore from now on, we will use MANET to refer both of them throughout the dissertation.

Nodes in the WSN have limited power supplies. Therefore WSNs require energy-efficient protocols and applications that would maximize the total lifetime of the WSN. Besides, nodes are prone to the failures, which would change the topology of the WSN unpredictably.

In a WSN, communications among the nodes are not continuous. This is because of the fact that WSN provides data to the users either on demand or upon event detection. While not in the communication phase, nodes either switch to "idle" phase or "hibernate" phase and they turn off their radios. In fact, this helps nodes to save energy and increases the total lifetime of the WSN.

WSNs may be subjected to different kinds of attacks (intrusions) against their availability (Denial of Service - DoS attacks) and against the integrity, authenticity and confidentiality of the information that is transmitted, processed and stored on the nodes. Besides, in some applications (e.g. military), WSNs are deployed in hostile environments and are operated unattended way, increasing the risk of being captured and compromised. These necessitate the security to be considered as one of the key design criterion for WSNs.

2.1 Characteristic of Wireless Sensor Networks

Comparison of WSNs vs. MANETs: Basic differences between WSNs and MANETs are [34]:

- The number of sensor nodes in a WSN can be several orders of magnitude higher than the nodes in a MANET.
- Sensor nodes are densely deployed, and are stationary in most of the scenarios. Whereas in MANETs, nodes are most likely mobile and because of that they are not densely deployed.
- Sensor nodes are prone to failures due to harsh environments and energy constraints.
- The topology of a sensor network changes frequently due to the failures, but not as rapid as in the case of MANETs where the nodes are moving.
- Sensor nodes are limited in computation, memory, and power resources; compared to the powerful nodes of MANETs which are typically laptop or a PDA.
- Sensor nodes may not have a global identification, on the contrary, MANET nodes generally deployed with an IP address.
- WSNs are widely used in environmental and building monitoring in which sensor nodes retrieve information on an event and pass this information to the base station; whereas

MANETs are used in disaster relief operations and tactical operations in which multiple kinds of radio carrying devices (aircraft borne, sea craft borne, ground craft borne, ground personnel borne, etc.) communicate with each other.

These differences (especially constraints on WSNs) greatly affect the implementation of secure data transmission in WSNs. As an example, low powered radio transmission of sensor nodes makes the communication channel susceptible to DoS attacks. Contrary to MANETs, advanced anti-jamming techniques (such as frequency-hopping spread spectrum communication) and physical tamper proofing of nodes are quite impossible in a WSN due to the requirements of a greater design complexity and higher energy consumption.

Features of WSNs: WSN is a distributed system, which does not have any infrastructure support (no gateways or routers). WSN consists of low cost, small sized nodes which are mostly stationary. Nodes in a WSN are generally deployed in large-scale, so that they need the ability to self-organize for the sake of wireless communications in a multi-hop way. Nodes need to operate autonomously with a limited amount of resources, requiring power efficient communication strategies (i.e., sleep, hibernate, awake cycles).

Network Topology of WSNs: In general, *Hierarchical Topology* is used for WSNs in which the network is divided into clusters. Key points of hierarchical topology are;

- Each cluster consists of two types of nodes: cluster heads (CHs) and subordinate (member) nodes.
- In most of the cases, varying levels of computational power within WSN; CHs have more computational power.
- Sensors do not communicate each other directly, the CHs are the gateways in doing so.
- Data flows from sensor nodes to the CHs.

Hardware specifications of WSNs:

- ARM-7 microprocessor is widely used in today's WSN nodes, which is working in the milli-watt range.
- XBee and XBee-PRO IEEE 802.15.4 OEM RF modules are embedded solutions providing wireless end-point connectivity to devices. These modules use the IEEE 802.15.4 networking protocol for fast point-to-multipoint or peer-to-peer networking. They are designed

for high-throughput applications requiring low latency and predictable communication timing.

- ZigBit is a low-power, high-sensitivity 802.15.4/ZigBee module. ZigBit is based on the industry leading Atmel Z-link hardware platform. The powerful ATmega 1281v MCU features 128KB of flash memory and 8KB of RAM. The transceiver boasts -101dBm of Rx sensitivity and up to +3dBm of Tx power. A link budget of 104dB gives the ZigBit a much longer range than competitive modules with lower link budgets. ZigBit packs impressive functionality into less than a square inch of space and offers superior radio performance with exceptional ease of integration. The ZigBit module eliminates the need for costly and time-consuming RF development, and shortens time to market for a wide range of wireless applications.

- Libelium Wasp motes [35] use ZigBit technology for telecommunications.

- Memsic MicaZ motes [36] use IEEE 802.15 technology for telecommunications.

Constraints and Challenges of WSNs: Increasing deployment of WSN for different applications is due to its inherent advantageous characteristics: such as, self-configuration, multi-hop behavior, no single point of failure, autonomous behavior, infrastructure-less operation, ease of deployment, and low cost. However, the benefits and flexibility of WSNs inevitably introduce many design challenges and constraints.

Main constraints of WSNs inherent from their design are;

- Limited bandwidth of wireless links lead to lower QoS compared to wired links.

- Limited battery power (typically 2AA sized batteries).

- Limited bandwidth (low throughput).

- Limited memory.

Main challenges in designing algorithms (i.e., telecommunications, networking and security) for WSNs;

- Dynamic topology due to nodes' mobility leads to packet losses, network partition, and network instability due to frequent route disconnections.

- Broadcast nature of wireless link leads to unavoidable interference and thus causes packet errors.
- Heterogeneous nodes with different capabilities (e.g., air interfaces) create further challenges.
- Network connectivity depends on transmission power, nodes density, and dynamic topology.
- Network reliability and robustness depends on autonomous nodes' behavior, node density, network load, topology changes, and link disconnections.
- Network security is critical since wireless links are prone to eavesdropping.
- Network scalability presents a daunting challenge for QoS delivery (for example, throughput or delay guarantees, etc.), network management, and security.
- By its nature, WSNs communicate through open air. Therefore it is vulnerable to various kinds of attacks such as eavesdrop-ping, Denial of Service (DoS), man in the middle, etc.
- The sensor hardware is not reliable, not tamper proof and operates in an unattended environment, which makes it an open target for node capture attacks (physical attack).
- Secure deployment of new nodes to an existing WSN without the need of renewal of the keys throughout the old nodes is problematic.
- Revocation of misbehaving nodes from WSN is also a problem to be solved.
- Sensor nodes are battery powered devices, so energy consumption is very important. Since wireless communication spends much more energy than computing; in order to extend the lifetime of the WSN, any algorithm including communication of the nodes has to be optimized.
- Providing security with minimum load to the sensor nodes is the main challenge.

2.2 Applications of Wireless Sensor Networks

The main objective of any WSN application is to provide solution to challenging real world problems, such as detection and tracking the movement of troops on a battle field, monitoring

environmental pollutants, measuring traffic flow on the roads, and tracking the location of the patients in a hospital.

Military: Some of the potential military applications are; homeland security, target tracking and reconnaissance.

Civil: Some of the potential civil applications are; industrial sensing (and automation monitoring), habitat and other environmental monitoring, scientific data collection, building monitoring, object tracking, intrusion detection (burglar alarms), emergency response and disaster recovery, hazard and structural monitoring, traffic control, inventory management in factory environment and health care (medical).

Commercial: Some of the potential commercial applications are; smart home monitoring (patient or elder people monitoring - proximity control), smart grid and remote home management (power switches, lights, door locks, air conditioner, etc.). Especially, remote home management would be very helpful to elderly people, wireless connected motion detectors (to monitor whether they fall down), stove on/off (they may forget to turn off the stove), refrigerator on/off. Smart home monitoring would be very helpful to track patient behavior(to observe the side effects of the medicines).

2.3 Security Issues in Wireless Sensor Networks

Wireless sensor networks (WSNs) have promising network infrastructure for many military applications, such as battlefield surveillance and homeland security monitoring [37]. In those hostile tactical scenarios and important commercial applications, security mechanisms are necessary to protect WSNs from malicious attacks.

A WSN may have to scale up to thousands of sensor nodes; and at the same time it needs simple, flexible, and scalable security protocols. However, to design such security protocols is not an easy task. Higher-level security and less computation and communication over-head are contradictory requirements in the design of security protocols for WSNs. In most cases, a trade-off must be made between security and performance [38].

2.3.1 Cost of Security

Security is a risk management of balancing the loss from breaches against the costs of security, both of which are difficult to measure.

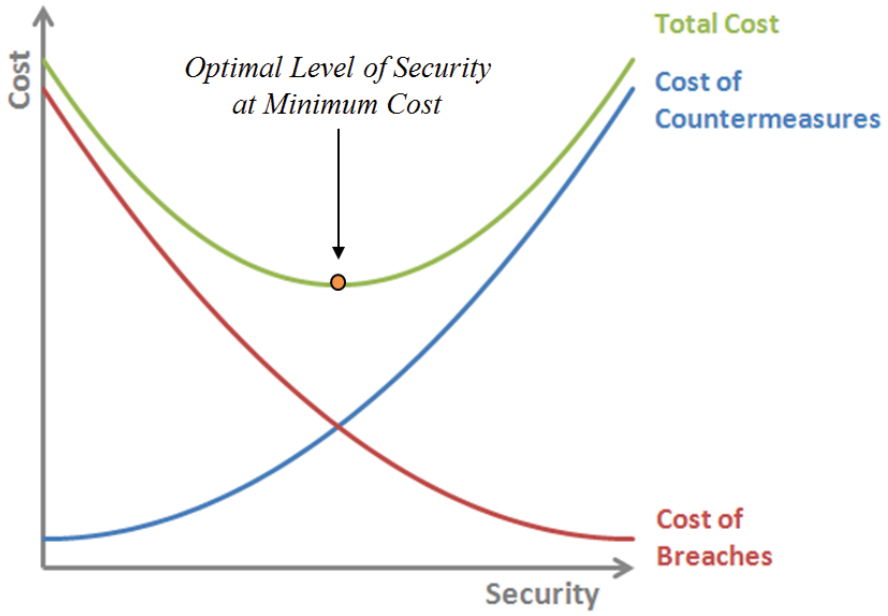

Figure 2.1 Optimization of security vs. cost [39].

From the optimization point of view to the information security and risk management; risk reduction is defined as the balance of the cost of the breaches against the cost of security counter measures to mitigate the risk. As shown in Figure 2.1, there is an optimal (minimum) point on the total cost curve (shown with orange dot) which can be found as the sum of the two costs, where the cost of countermeasures equals the cost of breaches. This is the point where optimal level of security is satisfied at minimum cost and should be the main target of the security administrators while designing their security plans.

2.3.2 Security Goals

Security goals for an information system are generally cumulated into three: CIA - Confidentiality, Integrity and Availability. Beside these trio, we may include non-repudiation and privacy as a supporting elements.

Confidentiality: Only authorized parties should be able to access the data provided by WSN. Attacks against confidentiality: Eavesdropping of wireless communications, physical capture of sensor nodes.

Integrity: If an authorized user receives data from the WSN, this data should be correct and valid; it shouldn't be changed by unauthorized parties. Attacks against integrity: Physical capture of some sensor nodes or intrusion of new malicious nodes.

Availability: WSN should always be able to answer any authorized request in its life time before the request expires. Attacks against availability: Denial of service (DoS) attacks such as power exhaustion attacks, jamming attacks, and collision attacks.

Non-repudiation: Neither the sender, nor the receiver can deny the transaction of the message.

2.3.3 Security Services

In order to fulfill the *Security Goals* in a network, *Security Services* are introduced:

Access Control: Access Control is about granting user access to network resources. It should provide access to legitimate users and deny access to illegitimate users. Access control is comprised of *authentication* and *authorization*.

Access control ensures that all accesses to objects (information resources) are authorized by regulating different privileged operations.

Precision agriculture is a good example for application of access control for WSNs [24]. For example let's say a WSN provider offers data services to subscribed farmers regarding information on their farms. Farmers may need to know the accurate readings on the humidity of the soil, in order to engage the sprinklers on time, before the crops be withered. In order WSN provider to make profit, only the legitimate users should get response to their queries from the WSN.

Authentication: It establishes a relation between a user and some identity (password, secret key, token, etc.). Authentication can be based on three techniques:

- Something the user knows, such as a password. Comes along with password management, which is required to prod users to change their passwords regularly, to select strong ones, and to protect them.
- Something the user possesses, such as a token. Each token has a unique secret cryptographic key stored within it, used to establish the token's identity via a challenge-response handshake.
- Something the user is, such as biometric data (finger prints, retina scan, etc.).

Technically, the best combination would be user-to-token biometric authentication, followed by mutual cryptographic authentication between the token and system services.

Authorization: Establishing a relation between a user and a set of privileges (access rights, allowed operations (read-write, read-only, etc.). This is generally implemented by access control lists.

Audit: It is the process of gathering data about activity in the system and analyzes it to discover security violations or diagnose their cause. Analysis can occur off-line after the fact or online in real time. In the latter case, the process is usually called *intrusion detection*. Audit has two components: the collection and organization of audit data, and an analysis of the data to discover or diagnose security violations.

2.3.4 Possible Attacks against WSNs

Appendix B summarizes attacks towards WSNs. The classification is provided according to the OSI protocol layer, meaning that attacks towards each OSI protocol layer (physical, data-link, etc.) are introduced separately.

2.3.5 Solutions to Defend against Attacks towards the WSNs

Appendix C summarizes solutions to defend WSNs against attacks towards them, especially the DoS attacks (blackhole, Sybil, flooding, etc.).

2.3.6 Patch Management

Patch management is a security practice designed to pro-actively prevent the exploitation of security vulnerabilities that exist within a network. The expected result is to reduce the time and money spent dealing with vulnerabilities and exploitation of those vulnerabilities. Timely patching of security issues is generally recognized as critical to maintaining the operational availability, confidentiality, and integrity of networks. However, failure to keep operating system and application software patched is one of the most common issues identified by security professionals. New patches are released daily, and it is often difficult for even experienced system administrators to keep abreast of all the new patches and ensure proper deployment in a timely manner.

Patch Management System (PMS)'s are useful to update patch and/or firmware to the end devices (such as sensor nodes) in order to defend the network against recently discovered/revealed

vulnerabilities. The importance of the PMSs has been arising, since the damages of the industrial espionage cases (e.g. Stuxnet virus [1]) revealed.

In the literature, several PMSs [40–43] are introduced to address the mentioned problems above. However they provide partial solutions to the problem and there is no "one size fits all" solution proposed yet. The following statement may summarize the challenge for developing PMSs: "There are too much vulnerabilities, requiring too many patches, with too many deployment mechanisms to be deployed to too many machines".

Readers, who are interested in application PMSs to Wireless Industrial Sensor Networks would find more information in the our paper [44].

2.3.7 Open Problems in WSN Security

Following is the list of open problems of WSN security.

- Trust: Trust is a big problem in WSNs. Especially following topics are worth to work on: trustworthiness, mutual trust, and trust management.
- Adaptability: Since WSNs possess frequently changing topology, security solutions must be highly adaptable.
- Scalability: Hence a WSN may consist of thousands of sensor nodes, the security mechanisms should be scalable.

Researchers can follow any topic on this list to conduct their research on WSN security.

2.4 Cryptography for Wireless Sensor Networks

2.4.1 Secret (Symmetric) Key Cryptography

Consider the WSN shown in Figure 2.2. By using Secret Key Cryptography (SKC), in order to send encrypted messages to its' neighboring sensor nodes (namely, S2, S3, S4 and S5), sensor node S1 needs to encrypt each message by using the pair wise "Secret Key" associated with that specific neighbor. In our example, S1 needs to encrypt the message outgoing to S2 with the pair wise Secret Key S1-S2, message outgoing to S3 with the pair wise Secret Key S1-S3, message outgoing to S4 with the pair wise Secret Key S1-S4 and finally, message outgoing to S5 with the pair wise Secret Key S1-S5. For the neighbors of S1 (namely, S2, S3, S4 and S5), in order to decrypt the messages coming from S1, they need to decrypt the encrypted message by using the same pair wise "Secret

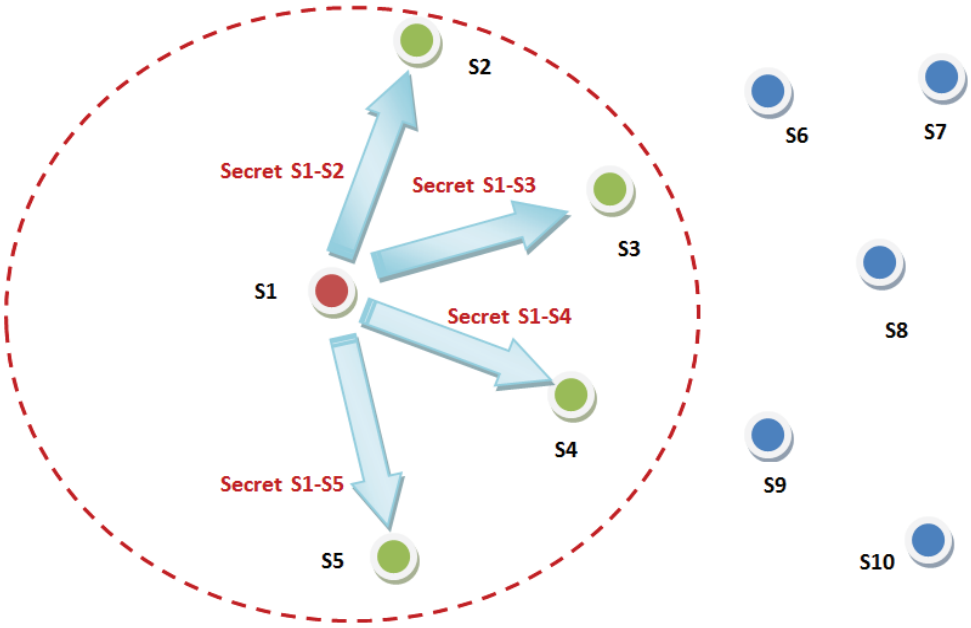

Figure 2.2 Illustration of secret key cryptography in wireless sensor networks.

Key" associated with S1 (Secret S1-S2, Secret S1-S3, Secret S1-S4, and Secret S1-S5.). By using this methodology, each node needs to store the pair wise secret keys associated with its neighboring nodes.

In a large network, distribution and management of these pair wise secret keys (Secret S_n-S_m) is a big problem in terms of communications overhead, memory usage, message complexity, and security resilience.

2.4.2 Public (Asymmetric) Key Cryptography

The computationally expensive portion of a Public Key Cryptography (PKC) system is typically the private key operations. PKC algorithms such as Rivest-Shamir-Adleman (RSA) algorithm, typically select the shorter keys as a public key in order to minimize the public key operations such as digital signature verification and encryption. Therefore, longer keys are selected as private keys, resulting with slow and resource demanding private key operations, such as decryption and signature generation. On the other hand, Elliptic Curve Cryptography (ECC) algorithm requires more overhead for encryption and signature verification than for decryption and signing [25]. Overall, the drawback of PKC is that it suffers from computational complexity (algorithms).

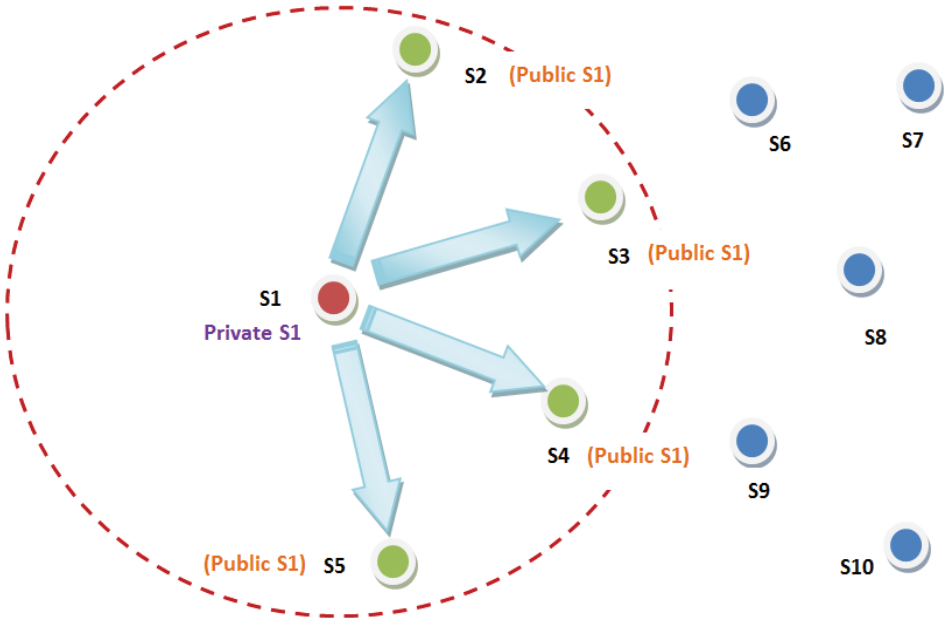

Figure 2.3 Illustration of public key cryptography in wireless sensor networks.

Consider the WSN shown in Figure 2.3. By using PKC, in order to send encrypted messages to its' neighboring sensor nodes (namely, S2, S3, S4 and S5), sensor node S1 just needs to encrypt message by using its "Private Key". For the neighbors of S1 (namely, S2, S3, S4 and S5), in order to decrypt the messages coming frm S1, they need to decrypt the encrypted message by using the "Public Key" of S1. By using this methodology, each node needs to store the public keys of its neighboring nodes as well as its own private key.

2.4.3 Hybrid Cryptography

When the number of the users in a WSN is too many, PKC algorithms are used for user authentication, as it scales much better than SKC algorithms. On the other hand, PKC algorithms require too much power to operate. Therefore for the communications between the sensor nodes, SKC algorithms are used. Sensor nodes in the communication range of the user serve as gateways between the two parts of the WSN which are using either PKC or SKC. The user communicates with the sensor nodes in the communication range using PKC, afterwards these sensor nodes communicate with the rest of the WSN using SKC as shown in Figure 2.4 [26].

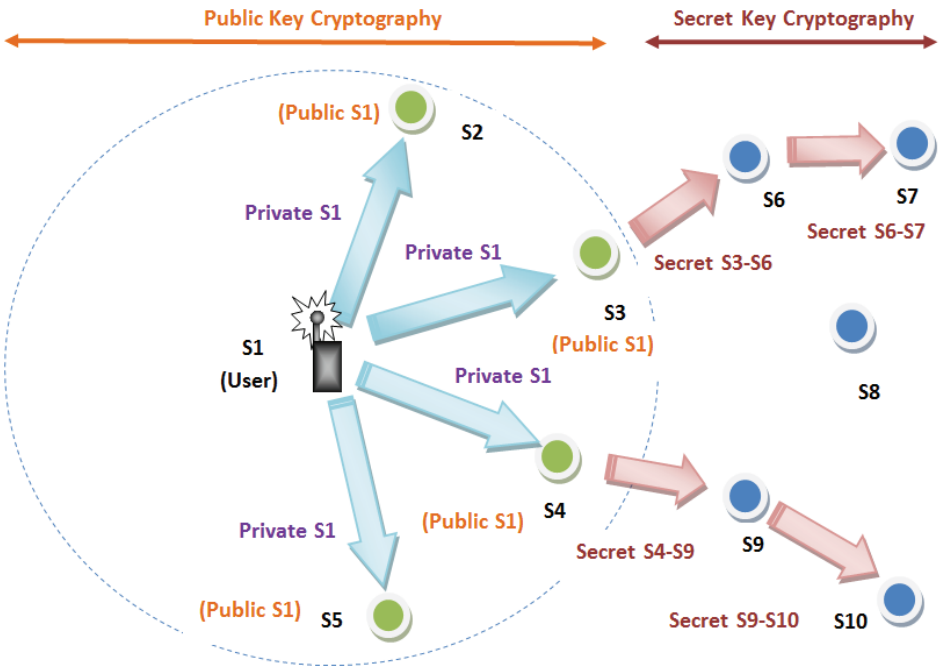

Figure 2.4 Illustration of hybrid cryptography in wireless sensor networks.

In hybrid cryptography approach, PKC is used for session key setup and authentication, whereas SKC is used to provide privacy. Here is the required steps, in order to achieve a secure network with hybrid cryptography:

1. Secure channel setup between the user and the WSN: The user executes a mutually authenticated key establishment protocol using PKC with some specified sensor nodes. The protocol results in the establishment of shared session keys between the user and each honest node which participated in the protocol run.

2. Authenticated querying: After the successful secure channel setup, the base station or the nodes in user's proximity forward user's queries into the sensor network and append to them some additional information enabling the other nodes to verify the legitimacy of the query.

The first phase naturally includes the user authentication phase. Although a secure channel is not required for access control, it is considered here because secure channel setup is a very well-studied standard procedure and incurs marginal additional costs in comparison to unilateral user

authentication. Additionally, secure channels between the users and the WSN are very likely to be required in the overall WSN design. For example, the answer to the query should be kept confidential. Moreover, the user should also be able to ascertain that it communicates with the genuine sensor network.

We briefly outline a possible solution to access control assuming that the query is addressed to a single sensor node, say sensor node S7. First, the user sends the query to the surrounding nodes (S2, S3, S4 and S5) using the previously established secure channels. Each node computes a message authentication code (MAC) on the query using the key shared with the node S7. For example, these keys could be computed using polynomial-based key pre-distribution. The computed MACs are sent back to the user who appends them to the query. The node S7 answers the query only if enough MACs are appended.

Note that in this solution, no coordination between the nodes in user's proximity is required. The node S7 answers the query only if enough MACs are appended to it. Such solutions should generally be preferred, as coordination requires additional messages, and therefore, additional resource consumption.

2.5 Security Provisioning Plan for Wireless Sensor Networks

As a starting point of a provisioning plan for any network security system (in order to set-up a rigid security system to cope with attacks), these steps should be followed:

1. Specification of the network resources.
2. Planning and design of intrusion prevention.
3. Planning and design of intrusion detection.
4. Planning and design of intrusion mitigation.

These steps are also visualized in Figure 2.5, as a flow chart of "Security provisioning plan for WSNs". In the flow chart, it is shown that, our security provisioning plan starts with "Specification of the network resources". This step is followed by the first line of defense, "Intrusion Prevention". Attacks, that would able to manage to pass the first line of defense, should be detected by the second line of defense, "Intrusion Detection". Finally, detected intrusions should be mitigated through the last step, "Intrusion Mitigation".

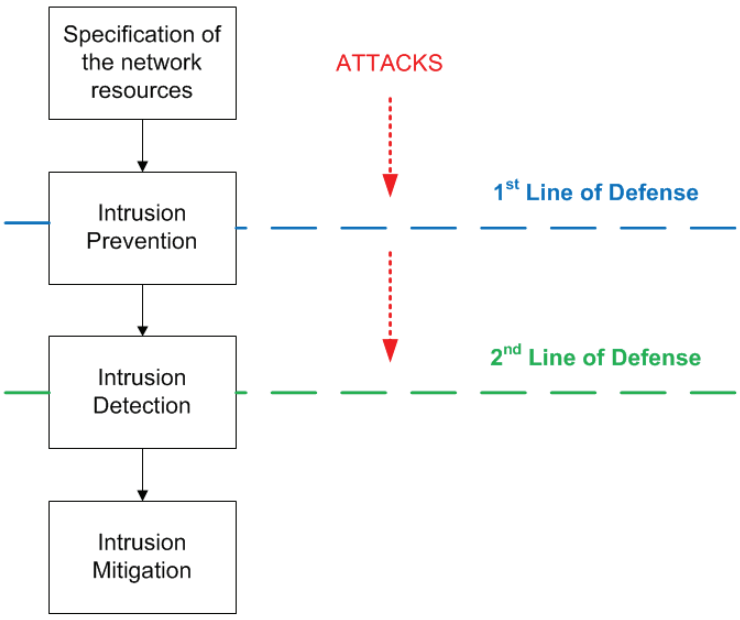

Figure 2.5 Security provisioning plan for wireless sensor networks.

2.5.1 Specification of the Network Resources

It's clear that specification of the network resources is the first step in any kind of network solution. We need to know what we need to keep secure from attackers and what kind of resources we have to fight against attackers. Besides, any solution that we devise has to be applicable to the network in concern. The topology (tree, ring, star, etc.) of the network has to be specified. Mobility (mobile, slightly mobile, stationary, etc.) of the network has to be specified. Hardware and software of the nodes need to be specified. Finally, frequency and data rate of the radio transmission need to be specified. All these specifications would affect our tailoring of the design from lightly to a moderate level.

2.5.2 Intrusion Prevention

Access control and authentication are the security measures to prevent intrusions. In any security system design, intrusion prevention constitutes the first line of the defense.

Intrusion prevention is the 2nd step in a network security plan. The network members should not be imitated by any non-member entity, and/or they should not be compromised by attackers, and/or their hardware should not be tampered by attackers, etc. In a wired network concept this is very well established. But in the case of wireless network concept, intrusion prevention is not as

easy as it is thought. Especially in WSNs, intrusion prevention is almost impossible considering the fact that physical capture of the nodes is almost inevitable (unless very specific tamper resistant hardware is used). Hence prevention of the intrusion is not feasible in WSNs, we will leave it as the final step of our security plan. Therefore, detection of the intrusion (misbehaving nodes) would be a good choice of starting point to a security plan.

2.5.3 Intrusion Detection

Intrusion is an unauthorized (unwanted) activity in a network that is either achieved passively (information gathering, eavesdropping) or actively (harmful packet forwarding, packet dropping, hole attacks, etc.). "Intrusion Detection" is detection of any suspicious behavior in a network performed by the network members. In our security plan, the 3rd step is the "Intrusion Detection" which would help mitigation step (4th step) by providing following information: identification of the intruder, and/or location of the intruder (single node/ regional), and/or time (date) of the intrusion, intrusion activity (active, passive), intrusion type (hole attacks worm hole, black hole, sink hole, selective forwarding attack, etc.), layer of the intrusion (physical, data link, network, etc.), etc. That's why "Intrusion Detection" is very important for a network from the security point of view.

In any security system design, intrusion detection constitutes the second line of the defense. intrusion detection systems are helpful to the overall security system in both ways:

- During any intrusion event, intrusion detection systems capture the logs of the event and alert the system administrators and/or the intrusion mitigation systems.
- The captured audits of an intrusion event can be investigated later on, in order to improve the first layer of defense (would provide enough clues about the gaps in the system), intrusion prevention.

Intrusion detection systems seek to help carry out audit controls. *Passive* intrusion detection systems analyze the audit data, usually offline, and bring possible intrusions or violations to the attention of the auditor. *Active* intrusion detection systems analyze audit data in real time and may take immediate protective response such as killing the suspected process and disabling the account. Two approaches used: ***Anomaly detection*** is based on the assumption that the exploitation of the vulnerabilities of the system involves abnormal use of the system. ***Misuse detection*** is based

on rules specific events, sequences of events, or observable properties of the system, symptomatic of violations.

2.5.4 Intrusion Mitigation

The 4th and final step of our security plan is the "Mitigation of the Intrusion". This step follows the intrusion detection and feeds (inputs) from the outputs of that step. That's why it is very important to design both steps in parallel and/or in coordination. The detected intrusions will be either by passed (zone blocking, re-routing), and/or ignored (packet dropping), and/or physically destroyed (in a military scenario), etc. according to the security strategy.

CHAPTER 3 :

INTRUSION PREVENTION WITH TWO LEVEL USER AUTHENTICATION IN HETEROGENEOUS WIRELESS SENSOR NETWORKS

3.1 Introduction

In any kind of network, there are two major steps to ensure security: intrusion prevention and intrusion detection. As a first line of defense, network is secured by using intrusion prevention methods such as authentication, authorization and access control. From computer security, we know that no system is completely secure unless it has no connection to outside world (close network). Hence, we are dealing with networking which means multiple of connections to outside world (open network), the intrusions are inevitable. Here, the second line of defense comes into the picture-intrusion detection: any kind of intruder that has managed to pass the first line of defense, need to be captured by this step. This chapter contributes to the first line of defense, namely to the intrusion prevention for heterogeneous (clustered) WSNs.

One of the major methods used for intrusion prevention is User Authentication (UA): If a user does not have enough credentials then (s)he will be denied to access the network. This would eventually prevent intrusions throughout the network, provided that the UA scheme is very well designed to cover entire network and leaving no weak points in terms of security. UA is critically needed for networks that are transferring confidential (sensitive and valuable) information to the legitimate users; such as the coordinates of a hostile vehicle for a military surveillance application, medical statistics of a patient for a health care application, etc.

Tactical WSN is a very good example of UA application. Let us say a WSN is deployed in warfare conditions and is used to gather tactical information of enemy forces on the war field. In this example, we will use the concept of proximity sensors that discover any vehicle or a soldier in their preset perimeter. Location of any hostile vehicle or soldier is very important under warfare and should be available to only friend forces to assess tactical advantage. Any friendly vehicle or a

soldier should not trigger an alarm in the proximity sensors (by using predefined communications using UA), but any existence of foe would do so. This is achieved by UA.

Health care is another example of UA application for WSNs. Let us say, a WSN offers instantaneous medical data service to subscribed health care employees such as doctors and nurses. Since the confidentiality of the data is important (i.e., privacy of patient medical records), only the legitimate users should get a response to their queries. Unauthorized users must be prevented from accessing the mentioned confidential information. Therefore UA is a must in these kinds of networks.

WSNs are characterized by unique characteristics:

- Severely constrained computational and energy resources: limited power supplies (limited energy), small memory sizes.
- Ad hoc operational environment: There is no structured network (there is no dedicated router or switch for network operations)and transmission bandwidth is narrow.

Therefore security techniques used in traditional networks cannot be adopted directly. As a result, although UA has been well studied for traditional networks, the models proposed for those networks cannot be applied directly to WSNs because of the unique characteristics that WSNs possess. UA in such a resource constrained WSN with minimum overhead provides significant challenges and is an ongoing area of research.

UA is very important for WSNs. In order to save the diminishing power resources, network should not be accessible by the unauthorized users. Any extra data transmission in the network generated by the malicious users (eg. flood messages) may cause battery power of a sensor node to be depleted faster. In a WSN, since an adversary can easily inject messages, any node receiving a message needs to make sure that the data used in any decision-making process originates from the correct source. UA prevents unauthorized parties from participating in the network: legitimate nodes should be able to detect messages from unauthorized nodes and reject them. UA is an intended feature that would prevent intruders and this way ensure trustability for WSN users.

In this chapter, we propose a secure and scalable UA scheme to prevent intrusions in WSNs. The rest of the chapter is organized as follows: Section 3.2 provides related work for UA in WSNs, motivation of this work and our research goals. Section 3.3 presents our proposed TLUA scheme. Section 3.4 provides the security analysis of TLUA scheme. Performance evaluation of TLUA scheme

is provided analytically in Section 3.5 and by simulations in Section 3.6. Finally, Section 3.7 concludes the chapter and outlines future work.

3.2 Related Work, Motivation and Research Goals

There are a couple of papers published in the area of authentication for WSNs: Perrig et al. [2] proposed a suite of security building blocks for WSNs called SPINS. It is optimized for resource constrained environments and wireless communication. SPINS has two secure building blocks: SNEP and μTESLA. μTESLA is a broadcast authentication scheme and SNEP provides two-party data authentication. TinySec [3] is a lightweight, generic security package that can be integrated into sensor network applications. It is incorporated into the official TinyOS release. LEAP [4] is a key management protocol for sensor networks that is designed to support in-network processing, while at the same time restricting the security impact of a node compromise to the immediate network neighborhood of the compromised node. LEAP also includes an efficient protocol for inter-node traffic authentication based on the use of one-way key chains. A salient feature of the authentication protocol is that it supports source authentication without precluding in-network processing and passive participation.

SPINS, TinySec, LEAP are not "User Authentication" schemes, they only provide one-hop "node authentication". It means that only two neighboring nodes authenticate to each other. End-to-end secure communication (i.e., user - sensor node) is not provided. In end-to-end secure communication, intermediate nodes cannot "see" what is being transmitted between two "end" parties, because of the encryption (two end parties share a common encryption key).

Recently, several schemes have been introduced as a UA scheme for WSNs: Wong *et al.* [5](throughout the chapter we call this as *WZCW scheme*) proposed a dynamic UA scheme for homogenous WSNs. Later this work was improved by Tseng *et al.* [6] (throughout the chapter we call this as *TJY scheme*) with the following advantages; including resistance of the replay and forgery attacks, reduction of user's password leakage risk, capability of a changeable password, and better efficiency. As discussed in [5], authors claim that weak-password authentication is not suitable for WSNs because it loads the computational overhead to the used cryptography algorithm. In other words the algorithm must be strong enough to compensate for the weakness in the key. Therefore they recommend strong-password authentication for WSNs in which computational load is light, owing to the

strength in the key. As a summary, both schemes use SKC for UA throughout the network which is not scalable for a large number of sensor nodes and users.

Although Benenson *et al.*'s scheme (throughout the chapter we call this as *BGR scheme*) [8] uses PKC for UA, it is not practical for WSNs because of the homogenous network structure, meaning that all the power and processing demanding PKC operations are supposed to be handled on the normal sensor nodes. As a result, authentication operations take minutes (as the authors of [8] confess) and batteries of the sensor nodes deplete faster (according to findings of [45]).

To the best of our knowledge, the only heterogeneous approach to the UA in WSNs in the literature is Le *et al.*'s [7] scheme (throughout the chapter we call this as *TTUA scheme*). In TTUA scheme, CHs are used as a backbone in the network so that the sensed data, after being collected, are transmitted through CHs towards the requesting users. For authentication purpose, SKC is issued between the CHs and the users. However, it is practically impossible to scale SKC keys to include a large number of users and sensor nodes, because of the memory limitations. Besides, in SKC, excluding existing users from the network and including new users to the network, requires key revoking and key re-distribution, which needs a considerable amount of communication overhead. These are the biggest constraints of the TTUA scheme.

The schemes mentioned above use either PKC approach (BGR) or SKC approach (WZCW, TJY and TTUA). Both approaches have advantages and disadvantages. PKC is preferable in terms of scalability and key management, but it is unsuitable for the sensor nodes due to higher processing power requirement and lower energy efficiency. In contrast, SKC is preferable in terms of lower processing power requirement and higher energy efficiency, but it is not scalable because of memory restrictions and it requires a complicated key pre-distribution, user revocation and key re-distribution.

WZCW, TJY, and BGR schemes are using homogenous WSN architecture, in which the network consists of one type of sensor node only. Nowadays, because of having better performance, heterogeneous WSN architecture is on demand. This kind of network consists of two types of nodes: Cluster Heads (CHs) and sensor nodes(s). TTUA scheme adopts heterogeneous WSN architecture and owing to the high processing powered CHs, it offers better performance compared to WZCW, TJY and BGR schemes. On the other hand, it is based on SKC just like as WZCW and TJY schemes. Therefore, it is not scalable for thousands of sensor nodes and users, occupies a significant

memory to store authentication codes. Thus, addition of new nodes and users is troublesome in terms of key distribution.

In this chapter, we propose a secure and scalable UA scheme, named as Two Level User Authentication (TLUA), to overcome mentioned shortcomings of the current state of the art schemes (namely WZCW, TTUA, TJY and BGR schemes). In our scheme, we adopt the idea of a two level heterogeneous network architecture in which a user communicates with a sensor node through Cluster Head (CH) of that sensor node. Our scheme uses Public Key Cryptography (PKC) between CHs and users, and Secret Key Cryptography (SKC) between CHs and sensor nodes. We have presented basics of our scheme in [27] and then presented our early findings of performance evaluations in [28].

This work extends our previous efforts in a more comprehensive, presentable and conclusive way; then evaluates our TLUA scheme and compare its performance with state of the art schemes in the literature (namely TTUA, TJY[1]and BGR schemes). Evaluations are provided in two ways:

1. Analysis on the following criteria are provided:
    - memory storage requirement,
    - scalability,
    - communication cost (in terms of time and energy),
    - computational cost.
2. Simulation on energy consumption and total delays are provided.

3.3 Two Level User Authentication Scheme

Part of our scheme is first presented in [27]. In this work, in order to relieve the confusion in the terminology (among the tiered networks and our two level architecture), we renamed our scheme as Two Level User Authentication Scheme (TLUA).

In our TLUA scheme, we adopted the idea of two level heterogeneous network architecture of TTUA scheme in which a user communicates with a sensor node through CH of that sensor node. Our proposed scheme not only keeps all the advantages of the TTUA scheme but also enhances its

[1]Since TJY scheme is a superior version of WZCW scheme, evaluations regarding TJY scheme will represent both schemes.

security by issuing PKC. Therefore, TLUA adopts (inherits) all the advantages of the PKC over SKC.

In [45], it is shown for WSNs that Elliptic Curve Cryptography (ECC) algorithm to have a significant advantage over Rivest-Shamir-Adleman (RSA) algorithm, as it reduces computation time and the amount of transmitted and stored data. Hence ECC is the best known algorithm in PKC [46, 47]; we adopt it to our TLUA scheme. By doing so, not only the scalability of the network is improved, but also security of the scheme is enhanced. In TLUA scheme, ECC is used for digital signature generation and verification between the users and the CHs; and Elliptic Curve Diffie Hellman (ECDH) key exchange protocol is used to exchange secret keys among CHs and sensor nodes.

3.3.1 System Model

In TLUA scheme, WSN consists of CHs and sensor nodes, representing a *Heterogeneous* network structure: 1) CHs have high processing capability and long lasting power supplies, such as iPAQ PDA [48]. 2) Sensor nodes have low processing capability and limited power supplies, such as MICA-2 motes [49].

CHs are assumed as trusted gateways to the sensor nodes. TLUA scheme takes advantage of high processing power CHs to decrease the processing load on the sensor nodes. Hence they have better power supplies compared to sensor nodes, and are capable to run power hungry PKC algorithms. Therefore, between CHs and users, a PKC algorithm (namely ECC) is used for UA purposes. Once a user is authenticated to a CH then allowed to access the sensor nodes through that CH. Since it is low power demanding, between CHs and sensor nodes an SKC algorithm is used for UA.

TLUA allows a user to register once and authenticate to the network many times. Users can also change the password anytime at will. We consider large WSN (100's of sensor nodes) deployed in any variety of environments. In our WSN's architecture, base station (BS) is the point of central control, which serves as a trusted key management facility. BS is many orders of magnitude more powerful than sensor nodes. Typically, BSs have enough battery power to surpass the lifetime of all sensor nodes, sufficient memory to store cryptographic keys, stronger processors, and means for communicating with outside networks. After the deployment, sensor nodes form groups, called clusters, see Figure 3.1. For each cluster, a powerful node (e.g. PDA) is assigned as a CH. CHs have higher communication power than sensor nodes and therefore possess far more radio transmission

coverage. CHs can communicate with each other and also with BS. In order to protect the keying materials, CHs are equipped with tamper-resistant hardware. This assumption is reasonable, hence the number of CHs in a heterogeneous WSN is relatively small (e.g., approximately 20-30 CHs for 1,000 sensors), and the cost of such tamper-resistant hardware is small [50]. Users are equipped with portable computing devices, such as laptops, with no power constraints compared to sensor nodes. Users interact with the WSN for data query and retrieval. After processing sensed information; the sensor node either sends the data upon event detection or stores it to serve for the next query.

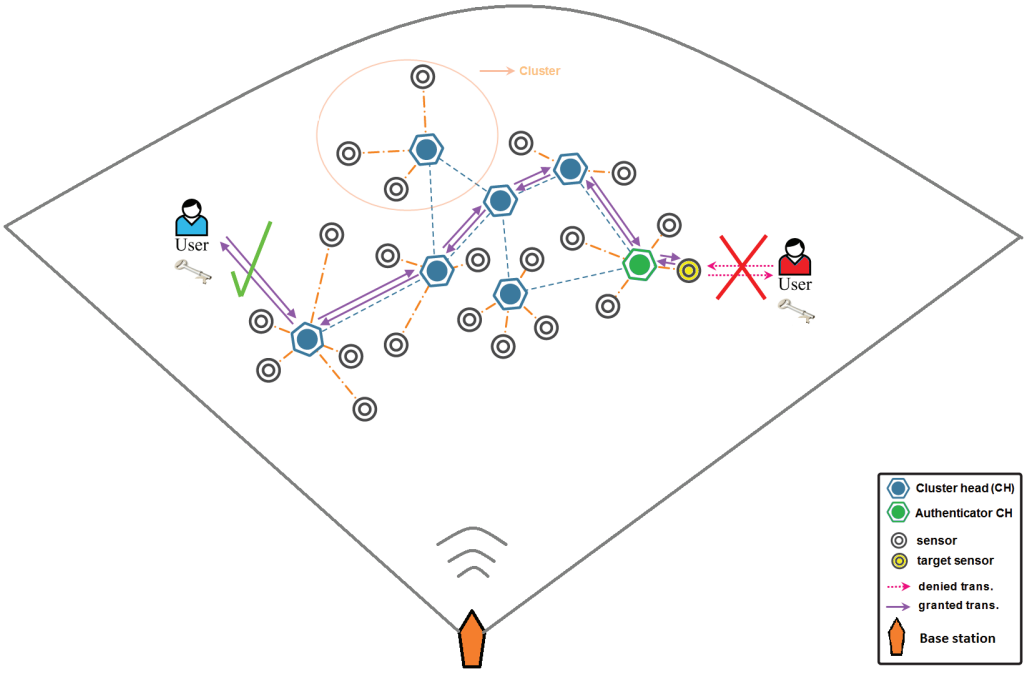

Figure 3.1 User authentication scenario in the TLUA scheme.

3.3.2 Key Agreement and Key Distribution

In our scheme we considered a Public Key Infrastructure (PKI) issuing ECC throughout the WSN. The network structure is the same as suggested in [7]. One BS serves as the certification authority for the network. ECC is used for encryption and decryption, Elliptic Curve Digital Signature Algorithm (ECDSA) is used for digital certificate generation and verification. The difference between a digital signature and a certificate is verification of a digital certificate reveals the content whilst verification of a digital signature reveals the hash of the content [51]. ECDH key agreement

protocol is used for key agreement between a CH (e.g. A) and their member sensor node(s), to be used as pair-wise MAC keys, $K_{A,s}$. Initially, BS generates elliptic curve parameters for ECC and ECDSA operations to be used by BS, CHs and users, and for ECDH operations to be used by CHs and sensor nodes. These parameters are; base point P, private key pri_key_{BS} and the corresponding public key $pub\_key_{BS} = pri\_key_{BS} \times P$ (where $\times$ stands for elliptic curve point multiplication) itself. BS also generates private-public key pairs for each sensor node (pri_key_s, $pub\_key_s = pri\_key_s \times P$) and for each CH (pri_key_{CH}, $pub\_key_{CH} = pri\_key_{CH} \times P$). Each sensor is pre-loaded with their private - public key pair and also the public key of the CHs. Each CH is pre-loaded with their private-public key pair and also the public keys of the sensor nodes.

In our scheme, between CHs and sensor nodes, in order to let both parties agree on a shared secret key, ECDH key agreement protocol is used as discussed in [25]. ECDH allows two parties to agree on the secret key of the SKC algorithm they are using: in our case it is MAC. In order to reduce energy consumption, all public keys needed for ECDH protocol are exchanged between sensor nodes and corresponding CHs before the deployment. Thus, no further communication is needed to exchange public keys. After deployment, each sensor node (s) computes a shared secret key ($K_{A,s}$) with its CH (e.g. A), for authentication purposes as follows:

- s computes the elliptic point R_s as shown in Equation 3.1:

$$R_s = (x_s, y_s) = pri\_key_s \times pub\_key_A \tag{3.1}$$

- A also computes another elliptic point R_A as shown in Equation 3.2:

$$R_A = (x_A, y_A) = pri\_key_A \times pub\_key_s \tag{3.2}$$

- Since Equation 3.3 holds, then $R_s = R_A$, and so does $x_s = x_A$. As a result $K_{A,s} = x_s$ is assigned as the shared secret key between s and A.

$$\begin{aligned} pri\_key_s \times pub\_key_A &= pri\_key_s \times pri\_key_A \times P. \\ &= pri\_key_A \times pri\_key_s \times P. \\ &= pri\_key_A \times pub\_key_s. \end{aligned} \tag{3.3}$$

Table 3.1 List of notations used in Section 3.3.

Abbreviation	Interpretation
A	cluster head named A
BS	base station
$cert$	certificate
$decrypt_x(y)$	decryption of y with key x, using ECC
ECC	elliptic curve cryptography
$ECDSA$	elliptic curve digital signature alg.
$encrypt_x(y)$	encryption of y with key x, using ECC
$H(.)$	hash value
ID_x	identification number of x
$K_{A,s}$	pair-wise key between A and s
MAC	message authentication code
pri_key	private key
pub_key	public key
s	sensor node
$sign_x(y)$	ECDSA signature
T	time stamp
U	user
$verify(z)$	verification of z
$\|$	concatenation
*	new

We assume that the key distribution between BS and CHs is established in a manner that all the CHs have the public key of the BS, namely pub_key_{BS}.

3.3.3 Authentication

TLUA includes three phases: Registration, Authentication, and Password Change. The operational functionality (handshake messages) of all these phases are summarized and illustrated in Figure 3.2[2].

User registration: User sends a request to the BS for registration to the WSN along with his ID encrypted with the public key of the BS, as shown in Equation 3.4:

$$user \rightarrow BS : \{Registration\_request; encrypt_{pub\_key_{BS}}(ID_U)\} \tag{3.4}$$

BS has the ID list of the legitimate users and provides each legitimate user a certificate. BS has private and public key pair $(pri\_key_{BS}, pub\_key_{BS})$ and the certificate is the user's ID signed

[2]For abbreviations and notations used in Figure 3.2, please refer to Table 3.1.

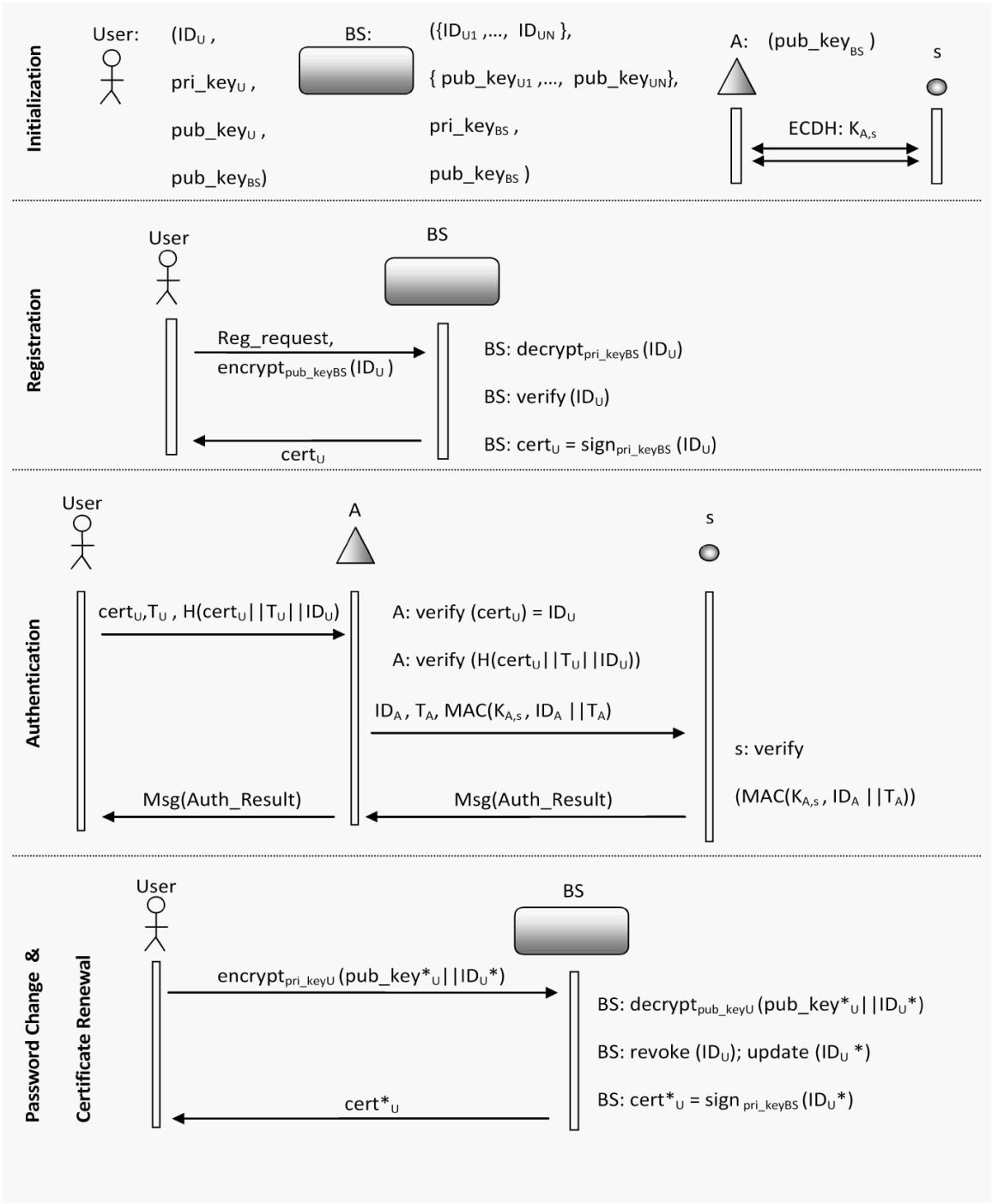

Figure 3.2 Communication handshake messages that are passed between different entities of the WSN for *Registration*, *Authentication* and *Certificate Renewal* phases of the TLUA scheme.

by the BS, using the private key (pri_key_{BS}). As a final step, BS sends back the certificate to the user (see Equation 3.5).

$$\begin{aligned} BS \quad &: \quad cert_U = sign_{pri\_key\_BS}(ID_U) \\ BS \quad &\rightarrow \quad User : cert_U \end{aligned} \tag{3.5}$$

In user authentication phase, with the public key of the BS (pub_key_{BS}), each CH can verify the certificate of the user and extract the ID of the user, namely ID_U.

User authentication: All the communications within the network are routed by the CHs. Let us consider the scenario where the user wants to access data aggregated at a sensor s (suppose A is CH of s), and let us also assume that A is the closest CH in the proximity of the user (intra communications and authentications among CHs are beyond the scope of our chapter). Then the authentication process includes the following steps:

- Step 1) The user sends his certificate $cert_U$ and time stamp T_U along with the hash value of those concatenated by user ID, ID_U to A as shown in Equation 3.6:

$$user \rightarrow A : cert_U, T_U, H(cert_U \| T_U \| ID_U) \tag{3.6}$$

  where $\|$ means concatenation and H stands for hashing algorithm such as SHA-1. In this representation, the hash value represents the variable (changes with the time, protected by time stamp) password of the user.

  Upon receiving an authentication request from the user, A first checks whether T_U is valid, if yes then it can verify the certificate of the user by using the public key of the BS (pub_key_{BS}) and extract the ID of the user, namely ID_U, as shown in Equation 3.7:

$$A : verify(cert_U) = ID_U \tag{3.7}$$

  Finally A verifies the hash value of the user by using the ID of the user as shown in Equation 3.8:

$$A : verify(H(cert_U \| T_U \| ID_U)) \tag{3.8}$$

- Step 2) If the verification is successful (meaning that the password provided by the user is correct), A sends s, its identification (ID_A) and time stamp (T_A) along with a MAC using its shared pair-wise key ($K_{A,s}$) with the sensor s, $MAC(K_{A,s}, ID_A \| T_A)$, as shown in Equation 3.9:

$$A \rightarrow s : ID_A, T_A, MAC(K_{A,s}, ID_A \| T_A) \tag{3.9}$$

Upon receiving the message, s first checks if T_A is valid. If yes, it verifies ID_A by generating a MAC with the shared pair-wise key with A ($K_{A,s}$) and comparing it with the received MAC, as shown in Equation 3.10:

$$s : verify(MAC(K_{A,s}, ID_A \| T_A)) \tag{3.10}$$

If all of these are successful, then the user is authentic. After successful authentication, sensor s is ready to send data to the user. s may send a short message to inform the user that he is authenticated via A.

User Password Change and Certificate Renewal: TLUA allows users to change their password by means of certificate renewal at their will. Users can do so through BS. The user encrypts the new public key ($pub\_key_U^*$) and its new ID ID_U^* with its current private key (pri_key_U), as shown in Equation 3.11:

$$user \rightarrow BS : encrypt_{pri\_key_U}(pub\_key_U^* \| ID_U^*) \tag{3.11}$$

After receiving the encrypted message, BS decrypts it by using the current public key of the user (pub_key_U), as shown in Equation 3.12:

$$BS : decrypt_{pub\_key_U}(pub\_key_U^* \| ID_U^*) \tag{3.12}$$

Then BS can sign the new ID (ID_U^*) by its private key (pri_key_{BS}) to obtain a new certificate ($cert_U^*$) and send it back to the user, as shown in Equation 3.13:

$$\begin{aligned} BS \quad &: \quad cert_U^* = sign_{BS}(ID_U^*) \\ BS \quad &\rightarrow \quad user : cert_U^* \end{aligned} \tag{3.13}$$

3.4 Security Analysis

In this section, we analyze the security of the TLUA scheme. In a two-party communication case, data authentication can be achieved through a purely symmetric mechanism: The sender and the receiver share a secret key to compute a Message Authentication Code (MAC) of all communicated data. When a message with a correct MAC arrives, the receiver knows that it must have been sent by the sender. In our TLUA scheme, MAC is used for all transmissions which involve sensor nodes and PKC (especially the ECC) is used in the backbone architecture of the network, namely between user side, BS and CH. Accordingly, not only the security aspect of the network is increased, but also most of the advantages of PKC and SKC are retained.

Authentication and encryption techniques can prevent an outsider to launch a *Sybil attack*[3] against WSN. However, an insider cannot be prevented from participating to the network. (S)he can achieve this by using the identities of the nodes (s)he has compromised. Besides, using globally shared keys allows an insider to masquerade as any node. PKC can prevent such an insider attack. It is one of the reasons to adopt it in our TLUA scheme. Although SKC is efficient in processing time for sensor networks, they generally require complicated key management, which needs large memory and communications overhead. On the other hand, PKC has simple key management with the more computational time trade-off. With the recent progress in ECC, Wang *et al.* [52] shows that PKC can be more advantageous than SKC not only with key management but also in terms of the memory usage and security resilience. This is another reason to adopt PKC to our TLUA scheme. In TLUA users can be added and revoked on the fly. CHs only need to keep the public key of the BS. Whereas in TTUA CHs need to keep the password list of the users every time.

[3]In Sybil attack, an adversary captures a single sensor node and illegitimately claims multiple identities to the sensor network.

In the TTUA scheme, the hash value of the user password is sent to the BS through a secure channel. Also, the hash values list of the CHs secret keys is sent from BS to the user through the secure channel. It means that in case of any intrusion into the secure channel, the WSN would be compromised. In our TLUA scheme, owing to the Public Key Infrastructure (PKI), a secure channel is not needed between user and BS. Therefore users do not have to plug in to the BS for key exchange. This means that users in our network are free to move anywhere but the coverage area of the BS.

In TTUA if the secret key (K_A) of the CH A is captured, then the network is compromised and all the user passwords stored on A must be revoked. In TTUA scheme, users change their passwords through CHs. In our TLUA scheme, users directly communicate with BS to change the password. Since CH is not involved in the password change session, TLUA is less vulnerable compared to TTUA. In TTUA scheme, a hash function (SHA-1) is used to secure the authentication message between the user and the CH. In our TLUA scheme, we use ECC which is more secure than SHA-1 given that both use same sized keys.

3.4.1 Node Compromising Attacks

Since CHs are equipped with tamper resistant hardware, it is impossible to compromise them. This way the SKC pair-wise keys between each sensor and associated CH is secured on the CH side. Also the PKC keying materials between CHs and users are also secured. The weakest element of our proposed scheme is the sensor nodes, since they do not have tamper resistant hardware. In terms of security, we do not let sensor nodes carry any valuable information to compromise the overall WSN. Hence the secret keys between a CH and each member sensor node are different, the furthest point any attacker can reach is the compromising of the link communication between the sensor node captured and the related CH. To defend this, in our TLUA scheme the secret keys between CHs and sensor nodes are updated at certain periods with ECDH protocol.

3.4.2 Replay Attacks

In the TLUA scheme an attacker cannot re-use the previous successful login message $H(cert_U \| T_U \| ID_U)$, because the time stamp T_U generated by the user protects this message to be used again after a certain time. After the useful time passes, CH will not allow access to the user. Thus, reply attacks are defended that way.

3.4.3 Impersonation Attacks

Our proposed scheme is resilient against impersonation attacks in the following manner: In authentication phase, an outsider tries to impersonate the login message $H(cert_U \| T_U \| ID_U)$ by fabricating ID_U as $ID_{U-guessed}$. The fabricated ID will change the hash value and will be caught by the CH throughout the hash value verification, as shown in Equation 3.14:

$$H(cert_U \| T_U \| ID_{U-guessed})! = H(cert_U \| T_U \| ID_U) \tag{3.14}$$

3.4.4 Brute-force Attacks

Our proposed scheme is resilient against brute force attacks in following manners:

- In the password change phase, an adversary intercepts the message $(encrypt_{pri\_key_U}(pub\_key_U^* \| ID_U^*))$ and tries to decrypt the message by estimating the public key of the user. So (s)he needs to try every combination of (pub_key_U) to decrypt the password change message. This kind of attack is known as brute force attack and is practically and cryptographically infeasible to be successful in useful time.

- In the password change phase, an adversary intercepts the message $(encrypt_{pri\_key_U}(pub\_key_U^* \| ID_U^*))$ and tries to estimate the private key of the user from the encrypted message, which is practically and cryptographically infeasible in useful time.

3.5 Performance Evaluation by Analysis

In this section, we analytically evaluate (by using theoretical calculations and also practical results from the literature) the performance of our proposed TLUA scheme and compare it to TTUA, TJY, and BGR schemes for the following criterions: storage requirement (memory), scalability, computational cost and communication overhead.

3.5.1 Storage

For Cluster Heads: TTUA scheme requires each CH to store user IDs and hashed password values, which adds up with the increasing number of users. As mentioned in [7], for the TTUA scheme, assuming that there are n number of users, user ID size is 8 $bytes$, and the hashed password value is 20 $bytes$, each CH has to store $n \times 28 + 120 bytes$ of data for the users. Whereas in our

Table 3.2 Comparison of memory storage (*bytes*) required on each sensor node and CH ($for 1,000 users$) in TLUA, TTUA, TJY and BGR schemes.

	TLUA	**TTUA**	**TJY**	**BGR**
CH	40	28,120	N/A	N/A
Sensor node	80	20	10,000	40

TLUA scheme, to authenticate the users, the only key that CHs have to store in their memory is the public key of the BS (pub_key_{BS}). This advantage is brought by the PKC. In our scheme, since we use 160 *bits* (20 *bytes*) elliptic curves, the public key size is 40 *bytes* (keep in mind that, for a 160 *bits* elliptic curve, certificate is 40 *bytes* long, public key is 40 *bytes* long and private key is 20 *bytes* long). Assuming n=1,000; the memory required to store keys on each CH are as shown in Table 3.2. Since TJY and BGR schemes do not require any CHs in their network, we will denote them as N/A in the table.

For Sensor Nodes: In TLUA scheme, each sensor node need to store private key of it self and public key of the CH for ECDH operation[4]. After the ECDH operation, CH and sensor nodes agree on a secret key $K_{A,s}$ which is 20 *bytes* long. As mentioned earlier, for ECDH, public key is 40 *bytes* long and private key is 20 *bytes* long. So, total memory space required for the keys are 80 *bytes* long. In TTUA scheme, each sensor stores a secret key $K_{A,s}$ which is 20 *bytes* long. In TJY scheme, every sensor node stores 10 *bytes* long key for each user. Therefore each sensor node needs to store $n \times 10 bytes$ long keying material. Finally, in BGR scheme, each sensor node need to store public key of the certification authority, which is 40 bytes. Assuming n=1,000; the memory required to store keys on the each sensor node is as shown in Table 3.2.

3.5.2 Scalability

As mentioned in the previous section, owing to the PKC approach, the memory space available on CHs in TLUA scheme does not change with the number of users. So we can state that there is no limit on the number of users. Literally speaking, TLUA scheme may manage thousands of users without any problem. Where as in TTUA scheme, memory space available on CHs is inversely proportional to the number of users. In TJY scheme, memory space available on sensor nodes is inversely proportional to the number of users, whereas in BGR scheme (owing to the PKC approach) it does not change with the number of users.

[4]This public key-private key pair is used by ECDH key agreement protocol to generate $K_{A,s}$ in the network initialization phase.

Table 3.3 Comparison of total number of users to be supported in TLUA, TTUA, TJY and BGR schemes.

TLUA	TTUA	TJY	BGR
> 10,000	< 100	> 10,000	< 200

Following the calculations from the previous section, if the memory size of each CH and sensor node for storing the keys is allocated as 2 $Kbytes$, then the number of users that would be supported in both TTUA and TLUA schemes are as shown in Table 3.3. Its apparent that TLUA and BGR schemes are very flexible and scalable compared to TTUA and TJY schemes in terms total number of users to be supported. Although according to our calculations there is no limit on the number of users for TLUA and BGR schemes, we limit this number to 10,000; which is reasonable for practical applications.

3.5.3 Computation

To compare the computational cost we have two comparison criterion: *time cost* and *energy cost*. We are interested on the operations running on CHs and sensor nodes but not interested in the operations running on the user devices and the base station. After we calculate the time cost of each scheme, we will calculate the energy cost of each scheme accordingly.

As a reference for our calculations, we used broad variety of reliable research results from the literature, especially papers on application of cryptography primitives over 8-bit CPU devices (namely Atmel ATmega microcontrollers) and hand held PDA devices (namely iPAQ). For the calculations involving sensor nodes, we referred to following research papers: [45, 46, 53–60]. For the calculations involving CHs, we referred to following research papers: [25, 61, 62]. For CH devices, we will consider iPAQ H3670 PDA. For sensor nodes, we will consider Berkeley's MICA2 motes[5].

Time cost: We define T_{MAC}, T_{SHA1}, T_{RC5}, T_{XOR} and T_{VER} as computational time cost of performing hash based message authentication code (CBC-MAC), hash function (SHA-1), symmetric encryption (RC5), XOR operation, and digital signature verification with ECDSA, respectively. Following this convention the computational time costs of TLUA, TTUA, TJY and BGR schemes are presented in Table 3.4. Since TJY and BGR schemes do not require any CHs in their network, we will denote their time cost as N/A in the table.

[5]Atmel ATmega microcontroller is the main chip on MICA2 motes.

Table 3.4 Comparison of computational time cost on each sensor node and CH in TLUA, TTUA, TJY and BGR schemes, provided as analytically.

Scheme	Phase	Cluster head	Sensor node
TLUA	reg.	0	0
	aut.	$1T_{VER} + 1T_{SHA1} + 1T_{MAC}$	$1T_{MAC}$
TTUA	reg.	$1T_{RC5} + 1T_{SHA1}$	0
	aut.	$1T_{SHA1} + 3T_{MAC}$	$1T_{MAC}$
TJY	reg.	N/A	$1T_{SHA1}$
	aut.	N/A	$2T_{SHA1} + 2T_{XOR}$
BGR	reg.	N/A	0
	aut.	N/A	$2T_{VER} + 1T_{SHA1}$

Table 3.5 Time spent on MICA2 motes (sensor nodes) for processing each security primitive.

Operation	Time
T_{SHA1}	4.91 ms
T_{MAC}	7.56 ms
T_{XOR}	≈ 0 ms
T_{VER}	3.27 sec

According to practical implementations on MICA2 motes (sensor nodes), the computational time required for each security primitive are as shown in Table 3.5.

In the case of BGR scheme, which is a PKC approach to UA in WSNs, authors [8] provided their experimental result as follows: Authentication takes 375 sec of time on a sensor node. Considering that the paper was published in 2005, we revised this number with latest findings in the literature [60, 63]. With recently discovered fast point multiplications, ECDSA signature verification costs as less as 3.27 sec. Literally speaking, in our analysis, we used up to date numbers in order to provide a fare comparison.

According to practical implementations on PDA devices (CHs) (i.e., iPAQ H3670), the energy spent for each security primitive are summarized in Table 3.6.

Table 3.6 Time spent on iPAQ PDA devices (CHs) for processing each security primitive.

Operation	Time
T_{SHA1}	10.13 μsec
T_{MAC}	15.47 μsec
T_{RC5}	10.53 μsec
T_{VER}	130.82 $msec$

Table 3.7 Comparison of computational time cost on each sensor node and CH in TLUA, TTUA, TJY and BGR schemes, provided as numerically.

Scheme	Phase	Cluster head	Sensor node	TOTAL
TLUA	reg.	0	0	0
	aut.	$130.85msec$	$7.56msec$	$138.41msec$
	sub-total	$130.85msec$	$7.56msec$	**138.41 msec**
TTUA	reg.	$20.66\mu sec$	0	$20.66\mu sec$
	aut.	$56.54\mu sec$	$7.56msec$	$7.62msec$
	sub-total	$77.2\mu sec$	$7.56msec$	**7.64 msec**
TJY	reg.	0	$4.91msec$	$4.91msec$
	aut.	0	$9.82msec$	$9.82msec$
	sub-total	0	$14.73msec$	**14.73 msec**
BGR	reg.	0	0	0
	aut.	0	$6.545sec$	$6.545sec$
	sub-total	0	$6.545sec$	**6.545 sec**

By using practical results of Table 3.5 and Table 3.6 we updated Table 3.4 as shown in Table 3.7. According to these results we see that TTUA is the fastest scheme and BGR is slowest (almost 1,000 fold slower). Although our TLUA scheme is using PKC, its performance results are very close to the SKC based schemes (TTUA and TJY) owing to the high speed processing capabilities of its CHs. To provide a better comparison, we plotted the total time cost (in $msec$) of each scheme as shown in Figure 3.3.

In our TLUA scheme, CHs are not involved in the registration phase, therefore the computation cost is zero. The authentication phase takes almost 138 milliseconds for TLUA scheme and 8 milliseconds for TTUA scheme. Which means that TLUA scheme is slower than (almost 15 fold slower) TTUA scheme for the authentication phase, which is expected. This is the trade off for changing cryptography approach from SKC to PKC. But keeping in mind that, BGR scheme requires 6.545 sec for the authentication phase, our scheme is almost 50 times faster owing to the high processing powered CHs.

Energy cost: As in the case of time cost calculations, we define E_{MAC}, E_{SHA1}, E_{RC5}, E_{XOR} and E_{VER} as computational energy cost of performing hash based message authentication code (HMAC), hash function (SHA-1), symmetric encryption (RC5), XOR operation, and digital signature verification with ECDSA, respectively. Following this convention the computational energy costs of TLUA, TTUA, TJY and BGR schemes are presented in Table 3.8. Since TJY and BGR

[7]Note that this figure is plotted in logarithmic scale.

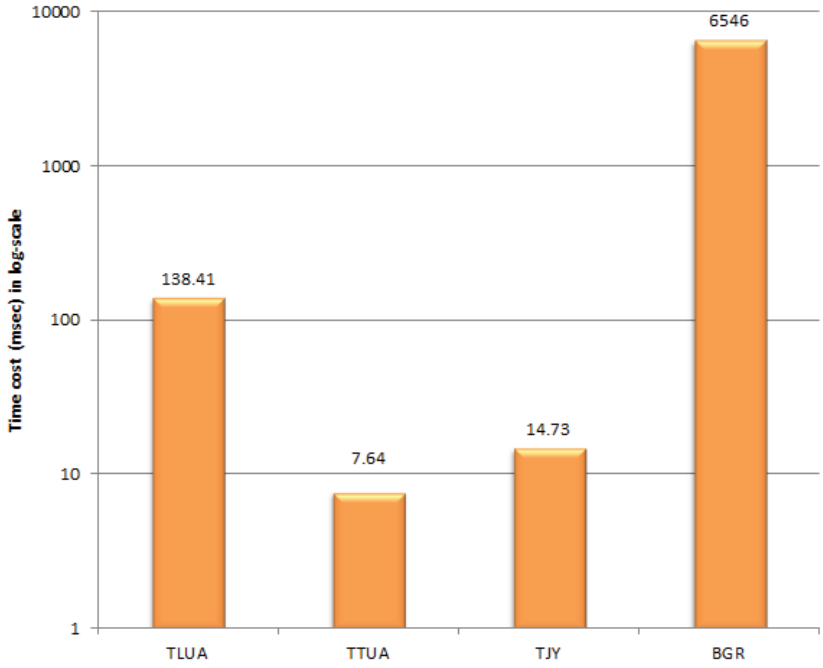

Figure 3.3 Comparison of total computational time costs ($CH + s$) of TLUA, TTUA, TJY and BGR schemes[7].

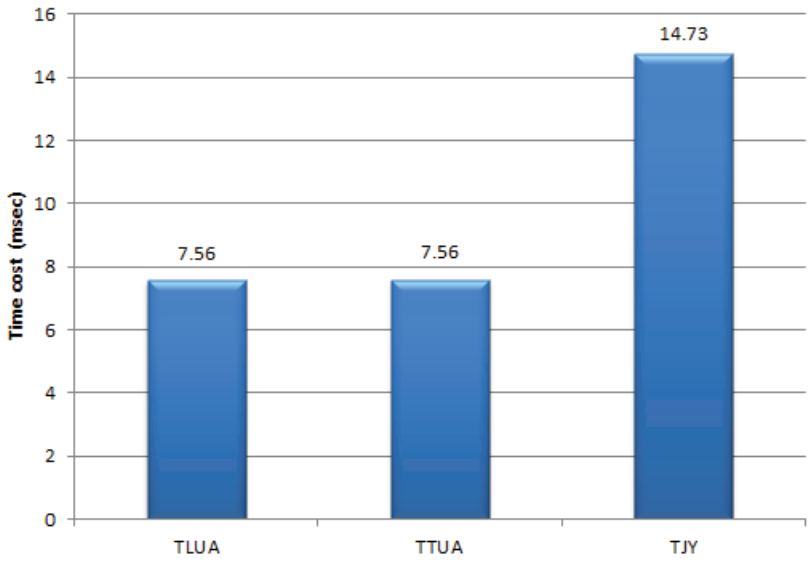

Figure 3.4 Comparison of computational time costs on sensor nodes of TLUA, TTUA and TJY schemes.

Table 3.8 Comparison of computational energy cost on each sensor node and CH in TLUA, TTUA, TJY and BGR schemes, provided as analytically.

Scheme	Phase	Cluster head	Sensor node
TLUA	reg.	0	0
	aut.	$1E_{VER} + 1E_{SHA1} + 1E_{MAC}$	$1E_{MAC}$
TTUA	reg.	$1E_{RC5} + 1E_{SHA1}$	0
	aut.	$1E_{SHA1} + 3E_{MAC}$	$1E_{MAC}$
TJY	reg.	N/A	$1E_{SHA1}$
	aut.	N/A	$2E_{SHA1} + 2E_{XOR}$
BGR	reg.	N/A	0
	aut.	N/A	$2E_{VER} + 1E_{SHA1}$

Table 3.9 Energy spent on MICA2 motes (sensor nodes) for processing each security primitive.

Operation	Energy
E_{SHA1}	$5.9\mu Ws/byte$
E_{MAC}	$9.0\mu Ws/byte$
E_{XOR}	$\approx 0\mu Ws/byte$
E_{VER}	$45.09mWs$

schemes do not require any CHs in their network, we will denote their energy cost as N/A in the table.

According to practical implementations on MICA2 motes (sensor nodes), the computational energy spent for each security primitive are as shown in Table 3.9.

According to practical implementations on PDA devices (CHs) (i.e., iPAQ H3670), the energy spent for each security primitive are summarized in Table 3.10.

By using practical results of Table 3.9 and Table 3.10 we updated Table 3.8 as shown in Table 3.11[8]. These results are very consistent with our findings for time cost calculations in previous section.

[8]Throughout these calculations we kept data size fixed as 20 *bytes*.

Table 3.10 Energy spent on iPAQ PDA devices (CHs) for processing each security primitive [61].

Operation	Energy
E_{SHA1}	0.76 $\mu Ws/byte$
E_{MAC}	1.16 $\mu Ws/byte$
E_{RC5}	0.79 $\mu Ws/byte$
E_{VER}	196.23 mWs

Table 3.11 Comparison of computational energy cost on each sensor node and CH in TLUA, TTUA, TJY and BGR schemes, provided as numerically.

Scheme	Phase	Cluster head	Sensor node	TOTAL
TLUA	reg.	0	0	0
	aut.	$196.27mJ$	$180\mu J$	$196.45mJ$
	sub-total	$196.27mJ$	**180 μJ**	$196.45mJ$
TTUA	reg.	$31\mu J$	0	$31\mu J$
	aut.	$84.8\mu J$	$180\mu J$	$269.6\mu J$
	sub-total	$115.8\mu J$	**180 μJ**	$295.8\mu J$
TJY	reg.	0	$118\mu J$	$118\mu J$
	aut.	0	$236\mu J$	$236\mu J$
	sub-total	0	**354 μJ**	$354\mu J$
BGR	reg.	0	0	0
	aut.	0	$90.298mJ$	$90.298mJ$
	sub-total	0	**90.298 mJ**	$90.298mJ$

Table 3.12 Comparison of communication cost for TLUA and TTUA schemes.

Phase	TLUA	TTUA
Registration	0	C_{br}
Authentication	$2C_{U-A} + 2C_{A-s}$	$2C_{U-A} + 2C_{A-s}$
Total	$2C_{U-A} + 2C_{A-s}$	$C_{br} + 2C_{U-A} + 2C_{A-s}$

According to these results we see that TTUA is the most energy efficient scheme and BGR is the worst (almost 300 fold more energy consumption). Although our TLUA scheme is using PKC, its performance results are very close to the SKC based schemes (TTUA and TJY) owing to the heterogeneous network architecture. To provide a better comparison, we plotted the total energy cost (in $microJoules$) of each scheme as shown in Figure 3.5.

In our TLUA scheme, CHs are not involved in the registration phase, therefore the energy cost is zero. The authentication phase spends almost 200 milliJoules for TLUA scheme and 300 microJoules for TTUA scheme. Which means that TLUA scheme spends more (almost 650 fold) energy than TTUA scheme for the authentication phase, which is expected. This is the trade off for changing cryptography approach from SKC to PKC. But keeping in mind that most ($> 99\%$) of this energy is spent on the CH. In our TLUA scheme, energy spent on the sensor node is same as the one on TTUA scheme, which is 180 microJoules. Compared to BGR scheme (which requires 90 milliJoules),TLUA scheme is very energy efficient (500 fold) on the sensor node.

[10]Note that this figure is plotted in logarithmic scale.

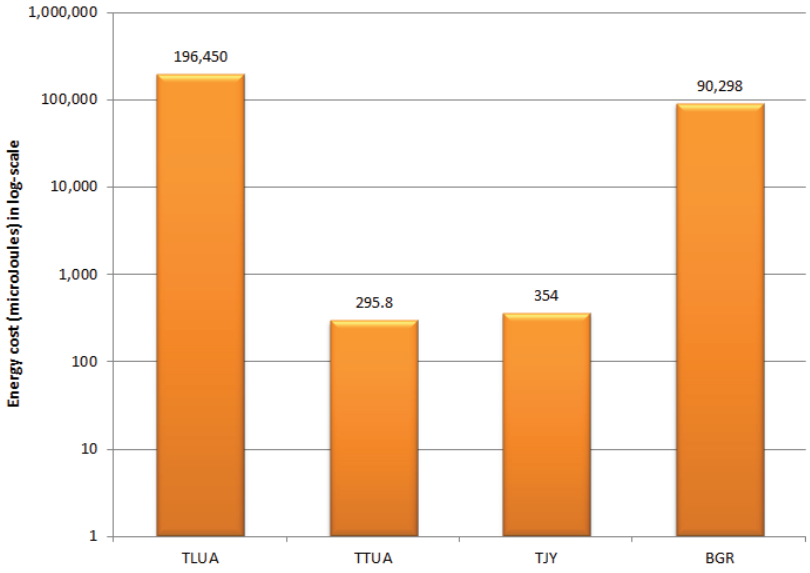

Figure 3.5 Comparison of total energy costs ($CH + s$) of TLUA, TTUA, TJY and BGR schemes[10].

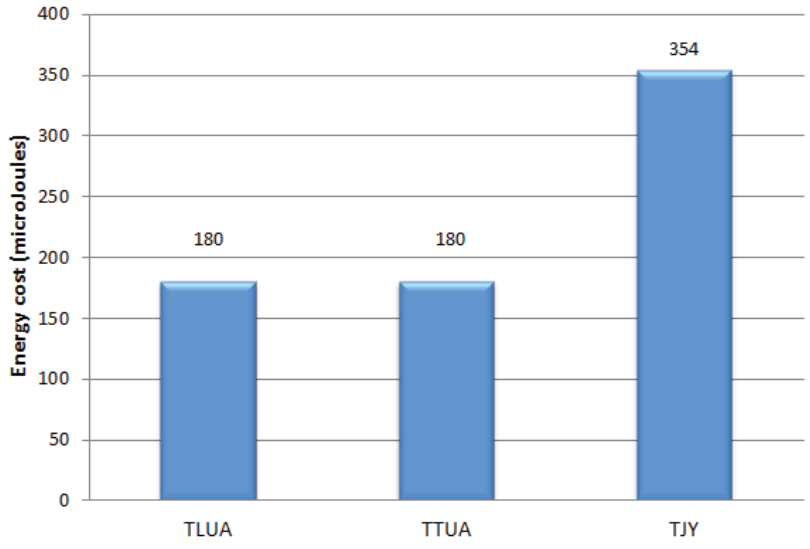

Figure 3.6 Comparison of energy costs on sensor nodes of TLUA, TTUA, and TJY schemes.

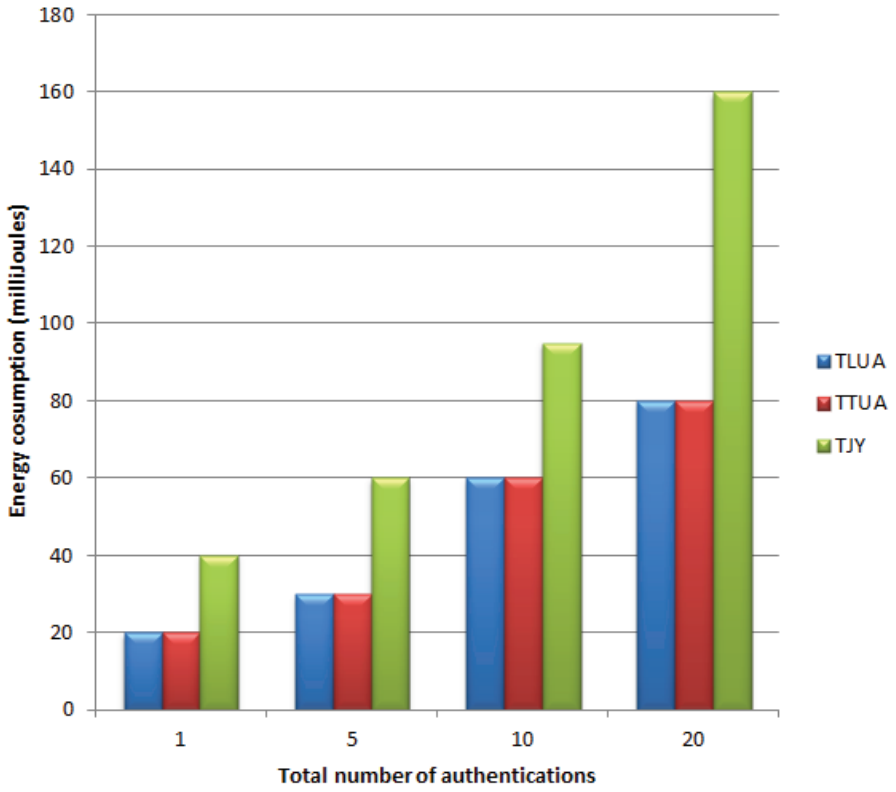

Figure 3.7 Comparison of energy consumptions on sensor nodes for three different schemes.

3.5.4 Communication

For communication cost, we are interested in the communications involving either CHs or sensor node s. To calculate communications cost, we define a number of notations as follows (all of these are in number of hops):

- C_{br}: Communication cost for broadcasting user ID and password to all CHs
- C_{U-A}: Communication cost between the user and the cluster head A
- C_{A-s}: Communication cost between cluster head A and sensor node s

For registration phase, TLUA has no cost on CHs or sensor nodes, whereas TTUA needs to broadcast user ID's and passwords to all CHs. For authentication phase, both schemes have the same cost, 2 messages sent between user and CH, and 2 messages sent between CH and sensor node. The communication costs of the both schemes are summarized in Table 3.12. This table provides the communications between; 1) the users and CHs, 2) CHs and sensor nodes (s).

According to the comparison of Table 3.12, we can conclude that both TLUA and TTUA schemes have same communication cost for the authentication phase. However, for the registration phase TTUA scheme requires a costly network-wide broadcast message, where as TLUA scheme requires none. So as a summary, our TLUA scheme outperforms TTUA scheme in terms of communications overhead.

3.6 Performance Evaluation by Simulation

We used SENSE (Sensor Network Simulator and Emulator) [64] to simulate and compare energy consumption and delay between TLUA [27], TTUA [7], and TJY [6] schemes. The simulation results show that the average energy consumption and delay time of different network topologies. Because cluster heads are much more powerful than sensor nodes, we only considered energy consumption of the sensor nodes. For each network topology, user's location and the login-node are randomly changed within the sensor field.

3.6.1 Simulation Model

The network deployment is similar to [50] with a BS and 300 sensors randomly distributed in a 300 $m \times 300$ m area. There are additional 20 CHs in the sensor field [50]. The transmission range of a sensor s and a CH is 60 m and 150 m, respectively. Sensors and CHs are formed in clusters. Each cluster has one CH. Sensors in the same cluster are connected with its CH via one or more hops. We use the same energy model used in ns-2.1b8a [65] that requires 0.66 W, 0.359 W, and 0.035 W for transmitting, receiving, and idling, respectively. We set the power consumption rate for SHA-1 and CBC-MAC calculation as 0.48 W according to [50] and [66]. As analyzed in [3] and [67], we set the time consumption for computing a CBC-MAC and a SHA-1 as 7.1 ms and 3.5 ms, respectively. The simulation uses MAC 802.11 Distributed Coordination Function (DCF). Two-ray ground is used as the radio propagation model. For routing in TLUA, TTUA and TJY schemes, we applied Ad hoc On-Demand Distance Vector (AODV) protocol. User ID length is 8 *bytes*, SHA-1 value is 20 *bytes*. As discussed in [3], the choice of 4 *bytes* MAC is not security detrimental in the context of sensor networks. So we applied 4 *bytes* CBC-MAC for every message and ran the simulation with five different network topologies. For each topology, five scenarios are applied, in which user's location and the login-node is randomly selected. For TJY scheme, we set the gate-way node in the center of the sensor field. We then averaged the results from those scenarios.

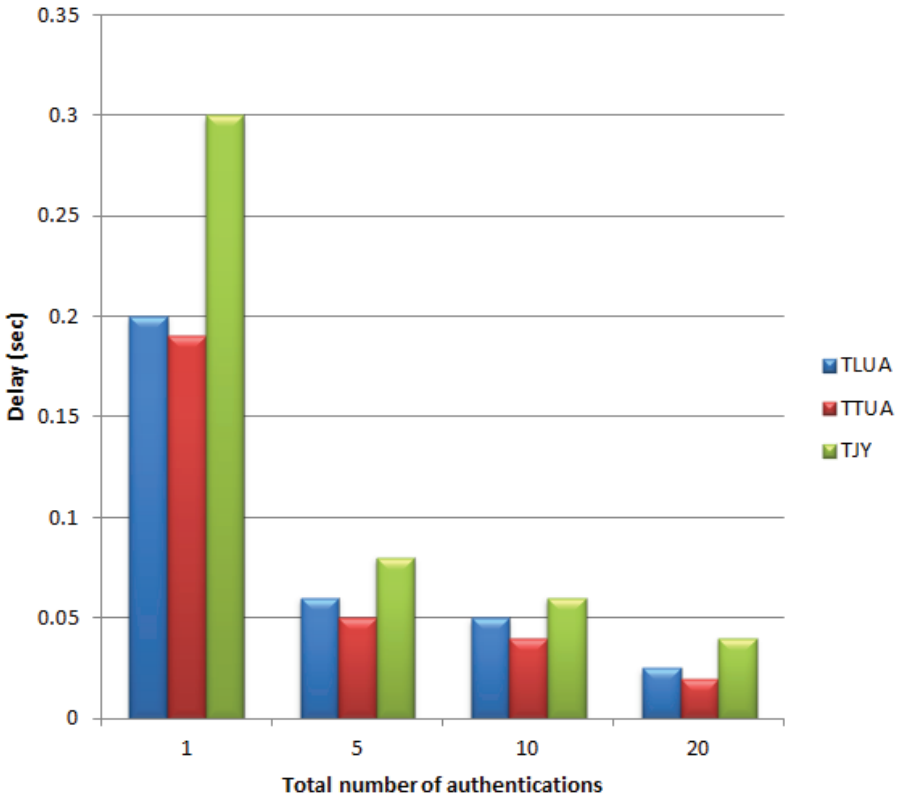

Figure 3.8 Comparison of computational times on the authentication phase for three different schemes.

3.6.2 Results

Our simulation results are shown in Figure 3.7 (this graphic compares total energy consumption on the sensor nodes for the authentication and registration phases) and Figure 3.8 (this graphic compares overall computational times for the authentication and registration phases). For one registration, the user is authenticated 1, 5, 10, and 20 times and in the graphs it is shown on the x-axis, respectively.

Figure 3.7 shows that the energy consumption (the energy consumption on sensor nodes for computation processes and for communication packets) of TLUA and TTUA is almost same and they are about half of TJY scheme. This is because computation cost of TLUA and TTUA are less than TJY scheme and they do not require any extra communication with the gate-way node during authentication process. However, TLUA and TTUA consume the same amount of energy because in

both schemes the communication cost between the user and targeted sensor, and the computational cost of the sensors are the same. This is consistent with our analytical results (See Figure 3.6).

Figure 3.8 shows that total delay time of TLUA is slightly greater than of TTUA but far less than TJY. This is consistent with our analytical results (See Figure 3.4). Although we used ECC signature verification in our scheme, this did not drop the overall performance significantly, owing to CHs with high processing speed (ECC signature verification takes about 1.65 ms on CH equipped with iPAQ [68]). 0.2 sec total delay of TLUA scheme is very compatible with TTUA scheme and way much better than TJY and BGR schemes. Furthermore, if the processing speed of the CH is increased (i.e., more powerful mobile devices), the delay on CH would be decreased dramatically, and our scheme would perform better than TTUA scheme.

3.7 Conclusions and Suggestions for Future Research

In this chapter, a novel IPS for heterogeneous WSNs, named as Two Level User Authentication (TLUA) scheme, is presented and then its performance is compared to the current state-of-the-art schemes in the literature. Proposed scheme employs both PKC and SKC approaches, so that it takes advantage of both schemes. Analysis and simulation results have shown that TLUA scheme is not only more secure and yet scalable than existing SKC based schemes, but also requires lesser processing power and provides higher energy efficiency than existing PKC based schemes. Proposed scheme brings advantages (scalability, flexibility) of PKC, without requirement of extra cost (in terms of energy) on the sensor nodes. Besides, time cost of the proposed scheme is very negligible compared to the other PKC based schemes (namely BGR scheme).

As a future work, hardware implementation (with real sensor devices) of the proposed TLUA scheme would be investigated.

CHAPTER 4 :

INTRUSION DETECTION SYSTEMS FOR WIRELESS SENSOR NETWORKS

4.1 Introduction

Owing to their easy and cheap deployment features, Wireless Sensor Networks (WSNs)[1]are applied to various fields of science and technology: To gather information regarding human activities and behavior, such as health care, military surveillance and reconnaissance, highway traffic; to monitor physical and environmental phenomena, such as ocean and wildlife, earthquake, pollution, wild fire, water quality; to monitor industrial sites, such as building safety, manufacturing machinery performance, and so on. [69]

On the other hand, security in WSNs is an important issue, especially if they have mission-critical tasks [70]. For instance, a confidential patient health record should not be released to third parties in a heath care application. Securing WSNs is critically important in tactical (military) applications where a security gap in the network would cause causalities of the friendly forces in a battlefield.

Solutions to security attacks against networks (wireless and/or wired) involve three main components [71]:

- Prevention (defense against attack): This step aims to 'prevent' any attack before it happens. Any proposed technique will have to defend against the targeted attack.
- Detection (being aware of the attack that is present): If an attacker manages to pass the measures taken by the 'prevention' step, then it means that there is a failure to defend against the attack. At this time, the security solution would immediately switch into the 'detection' phase of the attack in progress and specifically identify the nodes that are being compromised.

[1]See Appendix for the list of abbreviations used throughout this survey.

- Mitigation (reacting to the attack): The final step aims to 'mitigate' any attack after it happens by removing (revoking from the network routing tables) the affected nodes and securing the network.

Intrusion is an unauthorized (unwanted) activity in a network that is either achieved passively (e.g. information gathering, eavesdropping) or actively (e.g. harmful packet forwarding, packet dropping, hole attacks). In a security system, if the first line of defense, "Intrusion Prevention," does not prevent intrusions, then the second line of defense, "Intrusion Detection," comes into play. It is the detection of any suspicious behavior in a network performed by the network members. In any security plan, intrusion detection systems provide some or all of the following information to the other supportive systems: identification of the intruder, location of the intruder (e.g. single node or regional), time (e.g. date) of the intrusion, intrusion activity (e.g. active or passive), intrusion type (e.g. attacks such as worm hole, black hole, sink hole, selective forwarding, etc.), layer where the intrusion occurs (e.g. physical, data link, network). This information would be very helpful in mitigating (i.e., third line of defense) and remedying the result of attacks, since very specific information regarding the intruder is obtained. Therefore, intrusion detection systems are very important for network security.

WSNs have unique characteristics such as limited power supplies and energy, low transmission bandwidth, small memory size and data storage. Due to these restricted operating conditions (constrained computational and energy resources along with an ad hoc communication environment) of WSNs, most of the security techniques (including intrusion detection techniques) devised for traditional wired/wireless networks are not directly applicable to a WSN environment [24].

Designing an effective and efficient intrusion detection technique that is applicable to WSNs is a very big challenge, which motivated us to work on this research area. The first task of any research is to conduct an extensive literature review, which led us to the preparation of this survey as the first outcome of our research.

The rest of the chapter is organized as follows: In Section 4.2, a brief overview of IDSs, their classifications and their requirements is provided. Section 4.3 includes a brief survey of IDSs proposed for MANETs, followed by the comments regarding their applicability to WSNs. Section 4.4 specifies the challenges and restrictions of WSNs and highlights the differences compared to the other types of networks (wired/wireless). Then, a detailed literature review on IDSs devised for WSNs is provided along with comments on their prominent and lacking features. Finally, our paper is concluded by

comparing existing approaches, highlighting their lacking points and providing a general model for an IDS that would be applicable to WSNs.

4.2 Intrusion Detection Systems (IDSs)

In a network or a system, any kind of unauthorized or unapproved activities are called intrusions. An Intrusion Detection System (IDS) is a collection of the tools, methods, and resources to help identify, assess, and report intrusions. Intrusion detection is typically one part of an overall protection system that is installed around a system or device and it is not a stand-alone protection measure [72]. In [73], intrusion is defined as: "any set of actions that attempt to compromise the integrity, confidentiality, or availability of a resource" and intrusion prevention techniques (such as encryption, authentication, access control, secure routing, etc.) are presented as the first line of defense against intrusions. However, as in any kind of security system, intrusions cannot be totally prevented. The intrusion and compromise of a node leads to confidential information such as security keys being revealed to the intruders. This results in the failure of the preventive security mechanism. Therefore, IDSs are designed to reveal intrusions, before they can disclose the secured system resources. IDSs are always considered as a second wall of defense from the security point of view. IDSs are cyberspace equivalent of the burglar alarms that are being used in physical security systems today [74]. As mentioned in [73], the expected operational requirement of IDSs is given as: "low false positive rate, calculated as the percentage of normalcy variations detected as anomalies, and high true positive rate, calculated as the percentage of anomalies detected".

4.2.1 Requirements of IDSs

The IDS being designed, should satisfy following requirements:

- not introduce new weaknesses to the system,
- need little system resources and should not degrade overall system performance by introducing overheads,
- run continuously and remain transparent to the system and the users,
- use standards to be cooperative and open,
- be reliable and minimize false positives and false negatives in the detection phase.

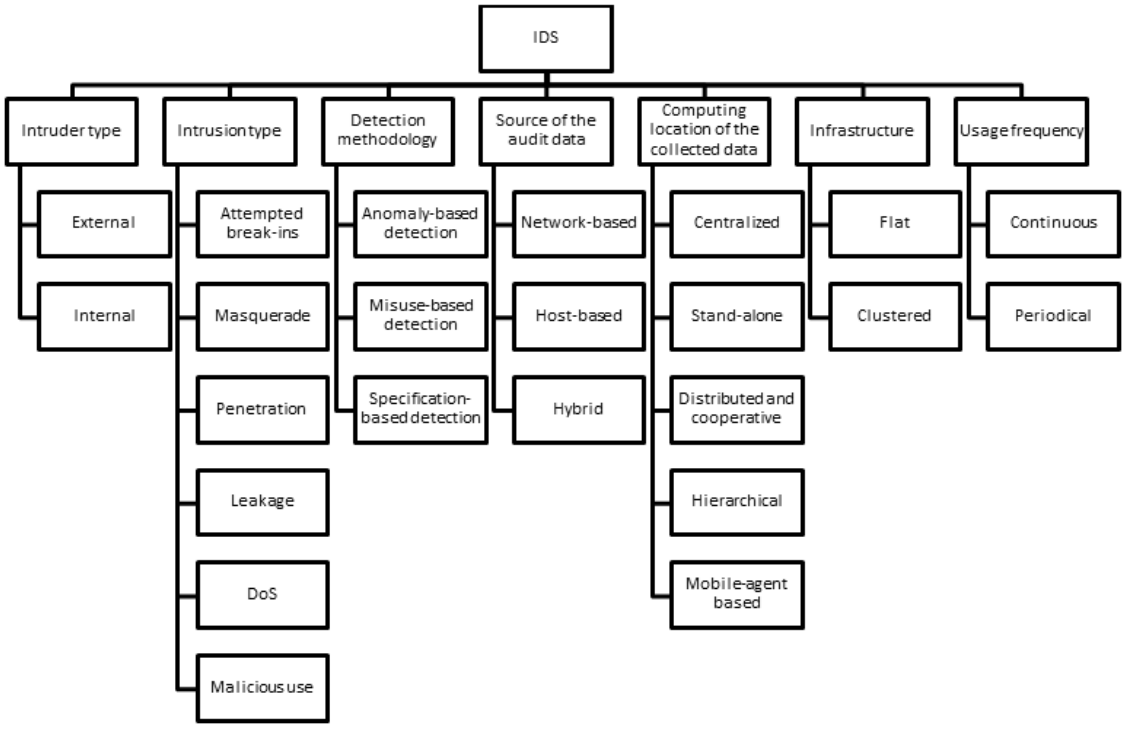

Figure 4.1 Classification of IDSs.

4.2.2 Classification of IDSs

As shown in Figure 4.1, IDSs can be classified as follows [75], [76], [77]:

Intruder Type: Intruders to a network can be classified into two types:

- An outsider using different means of attacks to reach the network.
- A compromised node that used to be a member of the network. According to [78], internal attacks against ad-hoc networks use two types of nodes:
  - Selfish node: Uses the network resources but does not cooperate, saving battery life for their own communications. It does not directly damage other nodes.
  - Malicious node: Aims at damaging other nodes by causing network Denial-of-Service (DoS) by partitioning, while saving battery life is not a priority.

An IDS can detect either external intruders, internal intruders, or both; according to its design aspects. But keeping in mind that internal intruders (insider attack) are not easy to detect, since

they have the necessary keying materials to neutralize any precautions taken by the authentication mechanisms.

Intrusion Type: Intrusions in a network may happen in various ways:

- Attempted break-in: An attempt to have an unauthorized access to the network.
- Masquerade: An attacker uses a fake identity to gain unauthorized access to the network.
- Penetration: The acquisition of unauthorized access to the network.
- Leakage: An undesirable information flow from the network.
- DoS: Blockage of the network resources (i.e., communication bandwidth) to the other users.
- Malicious use: Deliberately harming the network resources.

IDSs may provide partial detection solution to those attacks. But of course, all system administrators would like to have a perfect IDS that would able to detect all of the intrusions listed above.

Detection Methodologies: IDSs are functionally categorized into three groups: anomaly based detection, misuse based detection, and specification based detection:

- Anomaly Based Detection: This is based on statistical behavior modeling. Normal operations of the members are profiled and a certain amount of deviation from the normal behavior is flagged as an anomaly. The disadvantage of this detection type is that the normal profiles must be updated periodically, since the network behavior may change rapidly. This may increase the load on the resource constrained sensor nodes. According to [15], this model detects intrusions in a very accurate and consistent way (low false positive and false negative rates) under the condition that the network being observed follows static behavioral patterns. The advantage of this detection type is that it is well suited to detect unknown or previously not encountered attacks. According to Garcia-Teodoro *et al.* [79], anomaly based IDSs are further divided into three categories according to the nature of the processing involved in the behavioral model considered. These categories are modified according to [74] and the final categorization is illustrated in Figure 4.2:
  - Statistical based: In statistical based anomaly detection, the network traffic is captured and then a profile representing its stochastic behavior is generated.

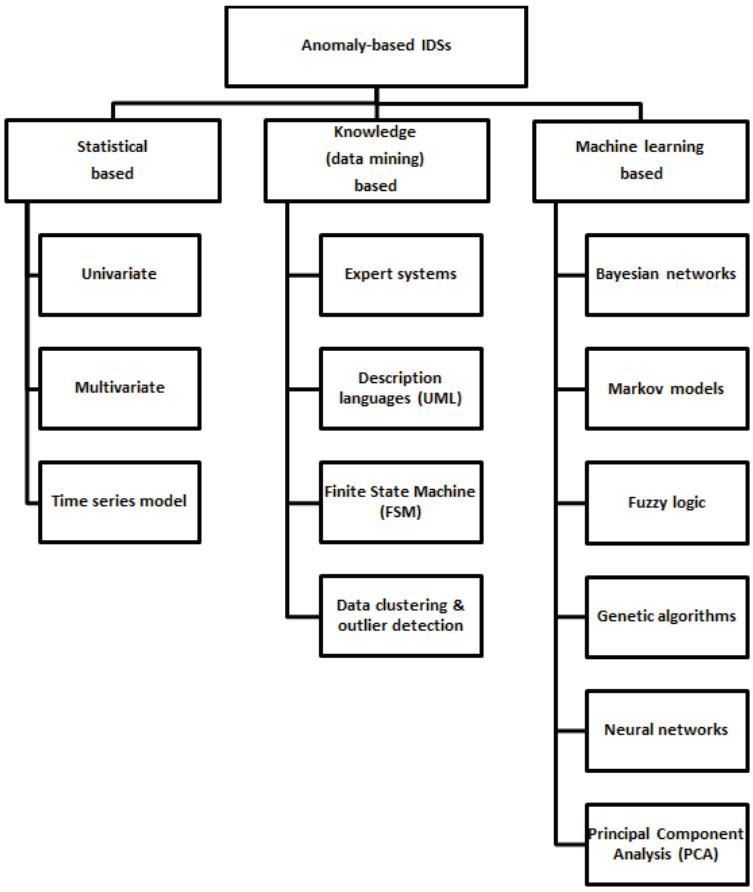

Figure 4.2 Classification of anomaly based IDSs according to their detection algorithms.

As the network operates in normal conditions (without any attack), a reference profile is created. After that, the network is monitored and profiles are generated periodically and an anomaly score is generated by comparing it to the reference profile. If the score passes a certain threshold, the the IDS will flag an occurrence of the anomaly.

Statistical based anomaly detection is divided in three subcategories:

* *Univariate:* Parameters are modeled as independent Gaussian random variables.
* *Multivariate:* Correlations between two or more metrics are also considered here.

* *Time series model:* Here, an interval timer is used along with an event counter that takes into account the order and inter-arrival times of the observations and also their values.

Statistical methods for anomaly detection are very well defined in [80] and here an example methodology for the detection of packet dropping attacks is summarized: Forwarding percentage (FP) of node m is the ratio of forwarded packets by m over the packets that are transmitted from M to m and m should forward, observed for a sufficient period of time (τ). It is calculated as follows:

$$\begin{aligned} FP_m &= \frac{packets\_actually\_forwarded}{packets\_to\_be\_forwarded} \\ &= \frac{\#(m,M) - \#([m],M)}{\#(M,m) - \#(M,[m])} \end{aligned} \tag{4.1}$$

Where:

* m: monitored node
* M: monitoring node
* #(m,M): the number of outgoing packets from m of which node M is the next hop
* #([m],M): the number of outgoing packets from m of which node m is the source and node M is the next hop
* #(M,m): the number of outgoing packets from M of which node m is the next hop
* #(M, [m]): the number of outgoing packets from M of which node m is the final destination
* FP_m: forwarding percentage of node m

If the denominator of equation (1) is not zero and if $FP_m = 0$, then this event is detected as "Unconditional Packet Dropping" and m is identified as attacker. If the denominator of equation (1) is not zero and if FP_m is less than a certain threshold (T_{FP}) and following condition (2) holds then this event is detected

as “Random Packet Dropping” and m is identified as attacker.

$$0 < FP_m < T_{FP} < 1 \tag{4.2}$$

– Knowledge based: Knowledge based anomaly detection rely on the availability of the prior knowledge (data) of the network parameters in normal operating condition as well as the one under certain attacks.

  * *Expert Systems:* It is based on rules classification of audit data.
  * *Description languages:* Diagrams (such as Unified Modeling Language (UML) diagrams) are generated based on the data specifications.
  * *Finite State Machine:* States and transitions are defined according to the available data set.
  * *Data clustering and outlier detection:* Observed data are grouped into clusters according to a specified similarity or distance measure. Points that do not belong to any cluster are named as the outliers.

– Machine learning based: In machine learning based anomaly detection, an explicit or implicit model of the analyzed patterns is generated. These models are updated periodically, in order to improve the intrusion detection performance on the basis of the previous results.

  * *Bayesian networks:* It is based on probabilistic relationships among the variables of interest.
  * *Markov models:* It is based on stochastic Markov theory in which the topology and capabilities of the system are modeled as states that are interconnected through certain transition probabilities.
  * *Fuzzy logic:* It is based on approximation and uncertainty.
  * *Genetic algorithms:* It is inspired by the evolutionary theory of biology.
  * *Neural networks:* It is based on the human brain foundations.

* *Principal Component Analysis (PCA):* Its is based on a dimensionality reduction technique known as PCA.

- Misuse Based (Signature Based or Rule Based) Detection: The signatures (profiles) of the previously known attacks are generated and are used as a reference to detect future attacks. For instance, a typical example of a signature would be: "there are 3 failed login attempts within 5 minutes" for the brute force password attack. The advantage of this type of detection is that it can accurately and efficiently detect known attacks; hence they have a low false positive rate. The disadvantage is that if the attack is a new kind (that was not profiled before), then the misuse detection would not able to catch it. Sobh [75] pointed out that these systems are very much like the anti-virus systems, which can detect most or all known attack patterns, but are of little use for the attack methods that are unknown yet. On the other hand, in [81], the authors present the following rules in order to monitor the network anomalies:
  - Interval rule: delay between the arrivals of two consecutive messages must be within certain limits.
  - Retransmission rule: the transit messages should be forwarded by the intermediate nodes.
  - Integrity rule: the original message from the sender must not deviate when it arrives to the receiver.
  - Delay rule: the retransmission of a message must occur after a certain wait time.
  - Repetition rule: same message can only be transmitted from the same node in certain number of counts.
  - Radio transmission range: the messages should be originated from the neighboring nodes only.
  - Jamming rule: the number of collisions for a packet transmission must be lower than a threshold.
- Specification Based Detection: A set of constraints that describe the correct operation of a program or protocol is defined. Then execution of the program with respect to the

defined constraints is monitored [76]. This methodology was introduced in [82], which provided the capability to detect previously unknown attacks, while exhibiting a low false positive alarm rate.

Sobh [75] identified the main distinction among the first two methods as: "anomaly detection systems try to detect the complement of bad behavior but misuse detection systems try to recognize known bad behavior".

Specification based intrusion detection techniques combine the advantages of both misuse and anomaly based detection techniques, by using manually developed specifications and constraints to characterize legitimate system behavior. Specification based intrusion detection techniques are similar to anomaly based detection techniques, in that both of them detect attacks as the deviations from a normal profile. Since specification based detection techniques are based on manually developed specifications and constraints, they have low false alarm rate compared to the high false alarm rated anomaly based detection techniques. On the other hand, the cost to achieve the mentioned low false alarm rate is that the development of detailed specifications and constraints would be very time consuming [83].

Source of the Audit Data: IDSs can be categorized into two groups according to the source of the audit data (depending on the location of the data to be analyzed):

- Host based Intrusion Detection System (HIDS): HIDS is concerned with the events on the host that they are serving. They are capable of (but not limited to) detecting the following intrusions: changes to critical system files on the host, repeated failure access attempts to the host, unusual process memory allocations, unusual CPU activity or I/O activity. HIDS achieves this by either monitoring the real-time system usage of the host or by examining log files on the host.

- Network based Intrusion Detection System (NIDS): NIDS passively or actively listens to the network transmissions, captures and examines packets that are being transmitted. NIDS can analyze an entire packet, payload within the packet, IP addresses or ports.

Computing Location of the Collected Data: IDSs are divided into four categories according to the computing location of the collected data:

- Centralized IDS: A centralized computer monitors all the activities in the network and detects intrusions by analyzing the monitored network activity data.

- Stand-alone IDS: An IDS runs on each node independently and every decision is based on the information collected at its own node. Members of the network are not aware of the intrusions happening around them because stand-alone IDS do not allow individual nodes to cooperate or share information among each other. They work as if they are alone.

- Distributed and Cooperative IDS: This is proposed for flat network infrastructures. Each node runs an IDS agent which participates (cooperatively participating in the global intrusion detection decisions and actions) in the intrusion detection and response of the overall network. If a node detects an intrusion with weak or inconclusive evidence, it can initiate a cooperative global intrusion detection procedure. If a node detects an intrusion locally with sufficient evidence, it can independently alert the network regarding an attack.

- Hierarchical IDS: This is proposed for multi-layer (clustering) network infrastructures. Cluster heads (CHs) are responsible for monitoring their member nodes, as well as participating in the global intrusion detection decisions.

- Mobile Agent based IDS: Each mobile agent is assigned to perform a specific task of the IDS on a selected node; and the intrusion detection is performed by the cooperative action of these selected nodes. After a certain time period or after a specific task is done, agents may relocate to other pre-defined nodes in order to increase network lifetime and/or efficiency of the IDS. Specifications of mobile agents are provided as follows:

    - Mobility: Mobile agent brings the code to the data on a remote host for asynchronous execution. This would help to reduce the amount of the exchanged data significantly.

    - Autonomy: Mobile agents are given a mission upon their creation: they should be capable of achieving their tasks without any external help.

    - Adaptability: Mobile agents should adapt their behaviors according to the information they gather while performing their tasks.

Infrastructure: Anantvalee *et al.* [76] divided IDSs (for MANETs) into two groups according to their network infrastructures:

- Flat: All nodes are considered as equal in capabilities and they may participate in routing functions. This infrastructure is suitable for civilian applications, such as networking in a classroom or a conference.

- Clustered: All nodes are not considered as equal. Nodes within transmission range are grouped into a cluster and they elect a node as cluster head (CH) to centralize routing information for that cluster. Generally, CHs consist of more powerful devices and backup batteries, resulting in a longer transmission range. Therefore, CH nodes form a virtual backbone of the network. Depending on the routing protocol, intermediate gateways may relay packets in between the CHs. This kind of infrastructure model is very suitable for military applications because of having a better command/control hierarchy.

Usage Frequency: According to the usage frequency, IDSs are divided into two categories:

- Continuous (on the fly): The IDS monitors the network continuously.

- Periodical: The IDS monitors the network in certain periods of time.

4.2.3 Decision Making in the IDS

There are two types of decision making mechanisms for IDSs:

- Collaborative decision making: All (or some) of the members of the network collaborate to conclude a decision regarding an event. For instance, in the case of majority voting, the final decision is made in favor of the majority of the members ending up with either of two decisions: "the event is an intrusion" or "the event is not an intrusion"

- Independent decision making: Each member concludes a decision regarding the events surrounding them.

According to [74], an IDS concludes either of four decisions (with non-zero probabilities) mentioned below as a result of the decision making process over an event:

- Intrusive but not anomalous (false-negative): There is an intrusion to the system, but the IDS fails to detect it and concludes the event as non-anomalous one.

- Not intrusive but anomalous (false-positive): There is no intrusion to the system, but the IDS mistakenly concludes a normal event as an anomalous one.

- Not intrusive and not anomalous (true-negative): There is no intrusion to the system, and the IDS concludes the event as non-anomalous one.
- Intrusive and anomalous (true-positive): There is an intrusion to the system, and the IDS concludes the event as an anomalous one.

For IDSs in WSNs, due to the nature of wireless communications, the following situations would result in false positives and that is why they need to be considered in the decision making model [78]:

- collisions
- packet drops
- limited transmission power
- fading battery power

4.2.4 Intrusion Response

When an attack is possible to happen, the IDS does not take preventive measures, since the prevention part is left to the Intrusion Prevention System (IPS). The IDS works in a reactive way compared to the proactive way of the IPS. Whenever the intrusion alert is generated by the IDS, the following action(s) would be taken according to the system specifications:

- An audit record should be generated.
- All the network members, the system administrator (if it exists) and the base station (if it exists) should be alerted about the intrusion. If possible, location and identity of the intruder should be provided in the alert message.
- If it exists, a mitigation method should be induced in order to stop the intrusion. For example, an automated corrective action should be generated through a collaborative action of the network members (especially the neighboring members to the incident).

4.2.5 Related Work and Suggested Readings

Readers, who are interested in the IDSs, can find more information (general information or specific areas other than WSNs) in the following papers:

- A very good classification of the IDSs is provided by Sobh [75].

- Classification of the IDSs for MANETs are provided by Ngadi *et al.* [72], Anantvalee and Wu [76], and Albers *et al.* [77].

- Garcia-Teodoro *et al.* [79], provided a survey of techniques, systems and challenges on the anomaly based NIDS.

- A brief survey of IDSs that are proposed for WSNs is provided in [84] and in contrast, our paper provides an extended survey with in-depth details comparing the proposed methods.

- A survey of IDSs for Collaborative systems is provided in [85]. A more specific survey on alert correlation in collaborative intelligent IDSs is presented in [86]. Another work on decentralized multi-dimensional alert correlation for collaborative IDSs is provided in [87].

- A survey of IDS in Cloud computing is provided in [88], which would be helpful to secure next generation networks.

- Garcia *et al.* [89] provides details of postmortem intrusion detection for Cyber security systems and computer forensics. They show a classifier method for analyzing log files by using hidden Markov model.

- Evasion techniques that are threatening IDS are presented in Cheng *et al.*'s work [90]. They provide details of 5 different techniques (DoS, packet splitting, duplicate insertion, payload mutation, shellcode mutation) and assess the effectiveness of these techniques on 3 most recent IDSs.

- Please note that the IDS that are investigated in this survey are related to information and computer security; and they are not related to the topic of "Intrusion detection for perimeter protection". Readers, who are interested in the later topic, please refer to the works presented in [91] and [92].

- Survey presented in this chapter does not include the methodologies and ideas that are proposed to secure the IDSs. Readers, who are interested in that topic may refer to Shakshuki *et al.*'s work [93].

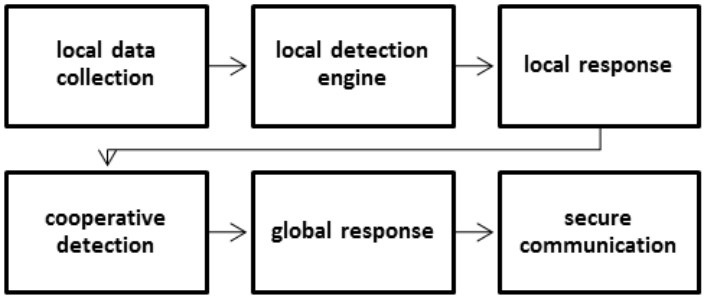

Figure 4.3 Building blocks of an IDS agent.

4.3 IDSs Proposed for MANETs and Their Applicability to WSNs

The IDSs for MANETs are very well investigated and here a summary of the literature is provided, in order to help the reader with a better understanding of the current state of the art. Following each review, we will discuss about each proposed IDS on the applicability to WSNs.

4.3.1 Agent Based Distributed and Collaborative IDSs

The first article on intrusion detection for MANETs was written by Zhang and Lee [94]. They proposed an agent based distributed and collaborative IDS which is compliant with the Wireless Ad Hoc Network operating conditions. As also mentioned in [76], the IDS agent described in [94] is composed of six blocks as shown in Figure 4.3: The local data collection block is responsible for collecting real-time audit data (user activities, system call activities, communication activities, and other traces) within its radio receiver range. This real-time audit data is analyzed by the local detection engine for the evidence of any kind of anomaly. In case of any anomaly detection, this block informs the local response and global response blocks (either one of them or both, depending the type of attack) in order to take a response against the anomaly (a possible intrusion). If the detection is inconclusive and needs more evidence, cooperation is conducted by the cooperative detection engine block and the communications with the neighboring agents needed for this cooperation is done through the secure communication block. For each agent, there is a module to detect anomalies, called the "local detection engine". These modules have two components, namely:

- features: describes a logical event in the network such as the percentage of the route changes of a node's routing table.

- modeling algorithm: uses features as an input to the rule based pattern matching algorithm and then specifies whether the incidence is a normal or not according to the predefined matching criterion.

In their model, every node participates in the decision making process. After a certain threshold, the local IDSs trigger the global IDS which necessitate collaborative decision of the nodes neighboring the flagged node. This decision is made through a majority voting process. Detection is made by using the means of "entropy": The higher the entropy, the higher is the probability of anomaly. The proposed method is useful to detect only the attacks against the routing protocols; i.e., mis-routing, false route updating, packet dropping, DoS.

After anomalies are detected, depending on the level of the anomaly, either a local response is created or a global (collaborative) response is created among with the neighboring nodes. And communications pertaining to this global response should be assessed through secure communication links among the nodes. According to the authors, determining the features that would lead the modeling algorithm to detect anomalies with low percentage of false positive detection rates is a non-trivial task.

The authors used two types of classifiers: Decision tree and Support Vector Machine. Updates of the routing tables are chosen as a trace data in three ways: percentage of the changed routes, percentage of changes in the sum of hops of all the routes, and the percentage of newly added routes. Trace analysis and anomaly detection are the two main methods for the IDS that are used by the authors. Data obtained from normal network routing operation is fed to the training algorithm to obtain reference values of the classifiers. Then deviations (correlate) from normal profile classifiers are used to determine the anomalies in the network routing.

The devised method was tested on the ns-2 simulator for the following MANET routing protocols: DSR (Dynamic Source Routing; a reactive, source initiated, on-demand routing protocol), AODV (Ad-hoc On-demand Distance Vector; a reactive, source initiated, on-demand routing protocol), and DSDV (Destination Sequenced Distance Vector; a proactive, table-driven, routing protocol). According to the results, their algorithm performs better for on-demand protocols than proactive protocols, because it is easier to observe the correlation between the traffic patterns and routing message flows in on-demand protocols.

As an extension to their previous work, Zhang *et al.* [73] introduced the idea of multi-layer integrated intrusion detection and response, which is built upon the distributed and collaborative

agent based IDS proposed in [94]. In the latest proposal, the intrusion detection module at each layer still needs to function properly, but detection on one layer can be initiated or aided by evidence from other layers. By this way, the authors claim that their IDS can achieve better performance in terms of both higher true positive and lower false positive detection rates. The proposed schemes [73, 94] might be applicable to WSNs in a sense that special care needs to be taken: As an example, they might be applied to a hierarchical WSN, where CHs might run the proposed schemes in a global sense and the sensor nodes in a local sense (division of the labor).

Following the works of Zhang *et al.* [73, 94], Albers *et al.* [77] improved the distributed IDS structure by including mobile agents with the design. Mobile agents bring the code to the data, as opposed to traditional approaches where data is conveyed towards the computation location. By this way, asynchronous execution of the agent is performed on a remote host. This decreases the amount of data traffic (involving the agents) in the network significantly. On the other hand, it increases the individual work load of each node, which is not desirable in WSNs. Besides, transmission of mobile code (an executable portion of the IDS is transferred to the nodes for on-site data processing) would decrease the bandwidth of the WSN, where bandwidth efficiency is of prime importance.

Kachirski and Guha [95] further improved the mobile agent notion of [77] by providing efficient distribution of mobile agents with specific IDS tasks (network monitoring, host monitoring, decision making and action taking) according to their functionality across the wireless ad hoc network. This way, the workload of the proposed IDS is distributed among the nodes to minimize the power consumption and IDS related processing times by all nodes. Therefore, this scheme is applicable to WSNs. Another improvement is to restrict computation-intensive analysis of overall network security to a few nodes only.

4.3.2 Clustering (Hierarchical) based IDSs

In Kachirski and Guha's approach [15], regular nodes do not participate in the global decision making process. Only the CHs are responsible for the global decision making process and the response. The main reason for this is to reduce the energy consumption. They wanted to conserve the energy of the majority of the nodes, by simply assigning them as subordinates under CHs.

In [80], clustering is used to select a single layer of sparsely positioned promiscuous monitors. These monitors are used to determine routing misbehavior via statistical anomaly detection. To conserve resources, a cluster based detection scheme is used in which a node is periodically selected

as the intrusion detection monitoring agent within each cluster. In the proposed architecture, a detection agent runs on each monitoring node to detect local intrusions and then it collaborates with other agents to investigate the source of intrusion and coordinate responses.

In [96], the authors proposed a scheme that applies decentralized, cooperative intrusion detection approach for clustered MANETs. Dynamic hierarchy is used as an organizational model which allows higher-layer nodes to selectively aggregate and reduce intrusion detection data as it is reported upward from the leaf nodes to a root. This infrastructure not only allows intrusion detection observations to be gathered efficiently from the network, but also provides incremental aggregation, detection, and correlation as well as efficient dissemination of intrusion response and management directives. The proposed scheme is tested for the following three scenarios:

- Intentional data packet dropping
- Attacks on MANET routing protocol
- Attacks on network and higher-layer protocols

Clustering based IDSs would be beneficial for WSNs if they are applied with special care. Because, CHs would deplete their energies faster than the other nodes which may cause segmentations (groups of nodes that are disconnected from each other) in the network. Therefore, extra batteries might be installed on CHs in order to help them to live longer, or CHs would be elected periodically in a sense that the node with the highest energy at each period would become the CH.

4.3.3 Statistical Detection based IDSs

Puttini *et al.* [97] provides an intrusion detection algorithm based on Bayesian classification criteria. Their design is based on statistical modeling of reference behavior using mixture models in order to cope with an observable traffic composed of a mixture of different traffic profiles due to different network applications. It is focused on the detection of packet flooding, an example of a DoS attack, and scanning of attacks against MANETs. The proposed model builds a behavioral model that takes into account multiple user profiles and uses a posteriori Bayesian classification of data as a part of the detection algorithm. In [32], the authors use estimated congestion at intermediate nodes to make decisions about malicious packet dropping behavior. They suggest that traffic transmission patterns should be used in concert with suboptimal MAC to preserve the statistical regularity from hop to hop. The proposed intrusion detection technique is a general one which is suitable

for networks that are not bandwidth limited but have strict security requirements such as tactical networks. Therefore it is not applicable to WSNs that have limited bandwidth. Statistical methods require too much data processing in order to sift the information that is valuable for statistics. Therefore, they are not applicable to WSNs.

4.3.4 Misuse Detection based IDS

Nadkarni and Mishra [98] proposed an IDS based on a misuse detection algorithm. Their implementation focused on distance-vector routing protocols such as DSDV protocol. Their implementation aimed at detecting DoS and replay attacks as well as compromised nodes. Their simulation results have provided significant results about not only the accuracy and robustness of the scheme but also the non-degradability of network performance. On the other hand, DSDV requires regular update for its routing tables which would not only deplete the energy resources of the nodes faster but also consume a portion of the valuable available bandwidth. Therefore, application of this algorithm to WSNs is not recommended.

4.3.5 Reputation based IDS

A reputation based IDS scheme promotes node cooperation through collaborative monitoring of the nodes and a grading system associated with the results of the collaborative monitoring.

Michiardi and Molva [78] used the concept of reputation in order to evaluate a member's contribution to the network. The higher a member's reputation, the more selected connections can be made with other members of the network. This means that, members of the network would rather communicate with that particular node compared to the lower reputation ones, which would encourage members to increase their reputations. The authors defined three types of reputations:

- Subjective reputation: evaluated considering the direct interaction between a subject and its neighbors.
- Indirect reputation: evaluated by the non-neighbor members of the community.
- Functional reputation: subjective and indirect reputations calculated with respect to different functions (packet forwarding, route discovery, etc.).

Their collaborative reputation evaluation system consists of two basic components:

- Reputation Table: A data structure, stored on each node which includes the reputation data pertaining to a node.

- Watchdog Mechanism: Calculates pre-defined functional reputations according to the data stored at the reputation table and then detects misbehaving nodes. Detection is based on a threshold value (e.g. zero) of the reputation; if the reputation of a specific member drops below the threshold value, then the watchdog mechanism will deny any communications with that member.

DoS attacks were also of concern to them. Therefore, they proposed a generic mechanism based on reputation to enforce cooperation among the nodes. Besides, this reputation mechanism prevents DoS attacks resulting from selfish nodes.

CONFIDANT protocol [99] works as an extension to reactive source routing protocols, such as DSR, and uses a reputation based system that rates nodes based on their malicious behavior. Alarm messages coming from other nodes are evaluated and the reputation of the node under investigation is updated only if the messages are coming from the fully trusted nodes. A neighborhood watching scheme is used to detect intrusive activity made by the next node on the source route. When a node detects a malicious neighbor, it sends an alarm message to other nodes on its list of trusted neighbors. The overall protocol may be summarized in one sentence as: "Cooperation of nodes for the sake of fairness".

Both of the proposed schemes of [78] and [99] are applicable to WSNs with a slight modification: The renewal period of the reputation tables would be decreased, in order to increase the bandwidth efficiency.

4.3.6 Zone based IDS

With Zone based IDS of Sun *et al.* [100], the network is divided into non-overlapping zones and each IDS agent broadcasts locally generated alerts inside the zone. Gateway zones are responsible for aggregation and correlation of locally generated alerts. Only gateway nodes can generate network wide alarms. Alerts indicate possible attacks and are generated by local IDS agents, while alarms indicate the final detection and can be generated only by gateway nodes.

The functionality of their proposed *local aggregation and correlation engine* is; to locally aggregate and correlate the detection results of detection engines. Whereas, the functionality of their proposed

global aggregation and correlation engine in gateway nodes is; to aggregate and correlate the detection results from local nodes in order to make final decisions.

Local alerts are generated according to two detection criteria: 1) Percentage of change in route entries, which represents the deleted and newly added routing entries in a certain time period; 2) Percentage of change in number of hops, which represents the change of the sum of hops of all routing entries in a certain time period.

According to the authors simulations (performed on GloMoSim network simulator); as the mobility decreased, their model responded with fewer false positives. Besides, aggregation algorithm of gateway nodes achieved much lower false positives than the IDS of local nodes, because they can collect information from a wider area and make more accurate decisions.

The proposed model detects intrusions in the routing layer of the OSI stack; it ignores other layers. Since the attacks happening in other layers would not be detected by this model, it is a partial IDS.

The proposed scheme requires each node to have the geographical information surrounding them. Although this is possible by attaching a global positioning system (GPS) receiver to the nodes in MANETs; it is infeasible in WSNs, because (most) sensor nodes are not generally equipped with GPS.

4.3.7 Game Theory based IDSs

In [101] and [102], the authors present a game-theoretic method to analyze intrusion detection in MANETs. They use game theory to model the interactions between the nodes of an ad hoc network. They model the interaction between an attacker and an individual node as a two player non-cooperative game. According to their assumptions, as long as the beliefs are consistent with the information obtained and the actions are optimal given the beliefs, the model is theoretically consistent.

The proposed schemes need a central processing unit, in order to process all the observations collected by the monitoring mechanism. This requires a high speed microprocessor as well as a large memory space to store the data to be processed. Therefore, in order to apply these schemes to WSNs, one should pick a centralized WSN, where a base station (BS) equipped with a computer that has high speed processing power and large memory. Besides, the schemes should be modified to decrease the traffic load in between each node and the BS. For example, a logging mechanism can

be used, where each node may store information regarding the data interactions with other nodes (and also if possible with the attackers). Then these logs may be sent to the BS, for the application of the game theory based detection.

4.3.8 Genetic Algorithm based IDS

Sen and Clark [103] investigated the use of evolutionary computation techniques to discover detectors suited to complex (lack of central computing unit, highly mobile nodes, limited resources) MANET environment. Authors applied grammatical evolution and genetic programming techniques to detect ad hoc flooding and route disruption attacks on AODV. Authors showed that their evolved programs performed good on simulated networks with varying mobility and traffic patterns.

Although this methodology might be very promising for MANETs where most of the nodes (e.g. PDAs) are powerful enough to run such energy consuming algorithms; it is not applicable to WSNs where sensor nodes have limited capacity on data processing along with the data storage.

4.3.9 Other Works

In [104], the watchdog mechanism is implemented on top of DSR protocol to verify that when a node forwards a packet, the next node in the path also forwards the packet; otherwise the next node is announced as misbehaving. Watchdogs run on each node, listens to transmissions of the neighboring nodes in a promiscuous mode. Watchdogs may not always be effective because of the packet collisions. The proposed watchdog mechanism is applicable to WSNs.

Wai *et al.* [105] proposed a hybrid IDS that can both work on wired networks as well as wireless ad hoc networks. The proposed model promises to use both anomaly and misuse detection algorithms. Both the details of the proposed model and the implementation results were not provided, thus making it impossible to compare its performance to the previously proposed models. Besides, the proposed scheme requires an end-to-end secure communication channel between nodes, which generally does not exist in WSNs.

MANETs became very useful for tactical networks such as command posts, vehicle convoys, autonomous robot systems, and also for infantry troops. The authors of MITE (MANET Intrusion Detection for Tactical Environments [106]) aim at developing prototypical solutions for intrusion detection in MANETs, especially in tactical scenarios. The results of MITE have been realized and evaluated as real-world implementations besides the simulation results. The authors proposed a

robust and resource saving sensor detector infrastructure as well as supporting components. The TOGBAD module of the proposed scheme uses a significant amount of the network traffic. Therefore, it is not applicable to WSNs, where the bandwidth is a scarce resource and needs to be utilized very efficiently.

Wei and Kim [107] used traffic prediction to detect intrusions in Wireless Industrial Networks. Authors proposed a data traffic prediction model based on autoregressive moving average (ARMA) using the time series data. According to their simulations, the model quickly and precisely predicted the network traffic and sifted out the attackers. Although the achievements seems promising; the proposed method brings extensive traffic load to the network for the sake of the monitoring data packets and also requires a centralized processing unit to store and analyze the whole traffic data, which are not provided in WSNs.

Readers, who are interested in IDSs designed for MANETs would find more information in the following papers:

- Brutch and Ko [82] provided a brief overview of research efforts on IDS for wired networks and wireless ad hoc networks. Besides, they provide classifications and different architectures of IDSs and highlight on their limitations in wireless ad hoc operation environment. They mention the methods to detect the attacks against the routing infrastructure and also methods to detect the attacks against mobile nodes.

- Mishra *et al.* [108] provided a brief introduction of MANETs and IDSs, and then summarized the key features of the IDSs proposed in the literature. They provided a survey on IDSs devised for MANETs.

- Sun *et al.* [83] provided a brief overview of intrusion detection techniques and a thorough survey on IDSs in MANETs. They also provided a literature overview of intrusion prevention algorithms proposed for WSNs. The article is written from the point view of secure in-network data aggregation.

- Sen and Clark [109], provided a survey of IDSs for MANETs. According to the authors, intrusion detection for MANETs is a complex and difficult task due to the dynamic nature of MANETs, their highly constrained nodes and the lack of central monitoring points.

- Ngadi *et al.* [72] also provided a brief survey of IDSs for MANETs.

Table 4.1 Proposed IDSs for MANETs and their applicability to WSNs.

Proposed system	Detection technique	Applicability to WSNs
Zhang and Lee [73, 94]	distributed and collaborative	applicable with modification
Albers *et al.* [77]	distributed and collaborative	not applicable
Michiardi and Molva [78]	reputation	applicable with modification
Kachirski and Guha [15]	clustering	applicable with modification
Kachirski and Guha [95]	distributed and collaborative	applicable
Huang and Lee [80]	clustering	applicable with modification
Sterne *et al.* [96]	clustering	applicable with modification
Puttini *et al.* [97]	statistical	not applicable
Rao and Kesidis [32]	statistical	not applicable
Nadkarni and Mishra [98]	misuse	not applicable
CONFIDANT protocol [99]	reputation	applicable with modification
Sun *et al.* [100]	zone based	not applicable
Patcha and Park [101, 102]	game theory	applicable with modification
Marti *et al.* [104]	watchdog	applicable
Wai *et al.* [105]	hybrid	not applicable
MITE protocol [106]	network monitoring	not applicable
Sen and Clark [103]	genetic algorithms	not applicable
Wei and Kim's [107]	autoregressive moving average	not applicable

4.3.10 Summary and Future Remarks

In this section, we present IDSs that are proposed for MANETs and discuss their applicability to WSNs. Some systems would be applicable directly (generic proposals), some would be applicable with major modifications, while the rest would not be applicable to WSNs (specific proposals), simply because of the unique design requirements of WSNs. Table 4.1 summarizes the schemes discussed so far, in terms of their detection technique and their applicability to WSNs.

Clustering (hierarchical networking) would be beneficial in adapting MANET IDS schemes to WSNs. For instance, consider the application of agent based IDS of [73] to a clustered WSN. The proposed IDS scheme would be divided into two categories as follows: Global IDS agents would be installed (with a full version of the scheme) on CHs; whereas local IDS agents would be installed (with a light version of the scheme excluding the global components) on each sensor node as shown in Figure 4.4. After two or more local IDS agents report the occurrence of an event, a global IDS agent would take charge and run a global detection sequence throughout the network. By running the full version of the scheme only on CHs and running the lighter version on the sensor nodes, the energy consumption of the whole scheme on the WSN would be significantly decreased and as a result; total life time of the network would be increased.

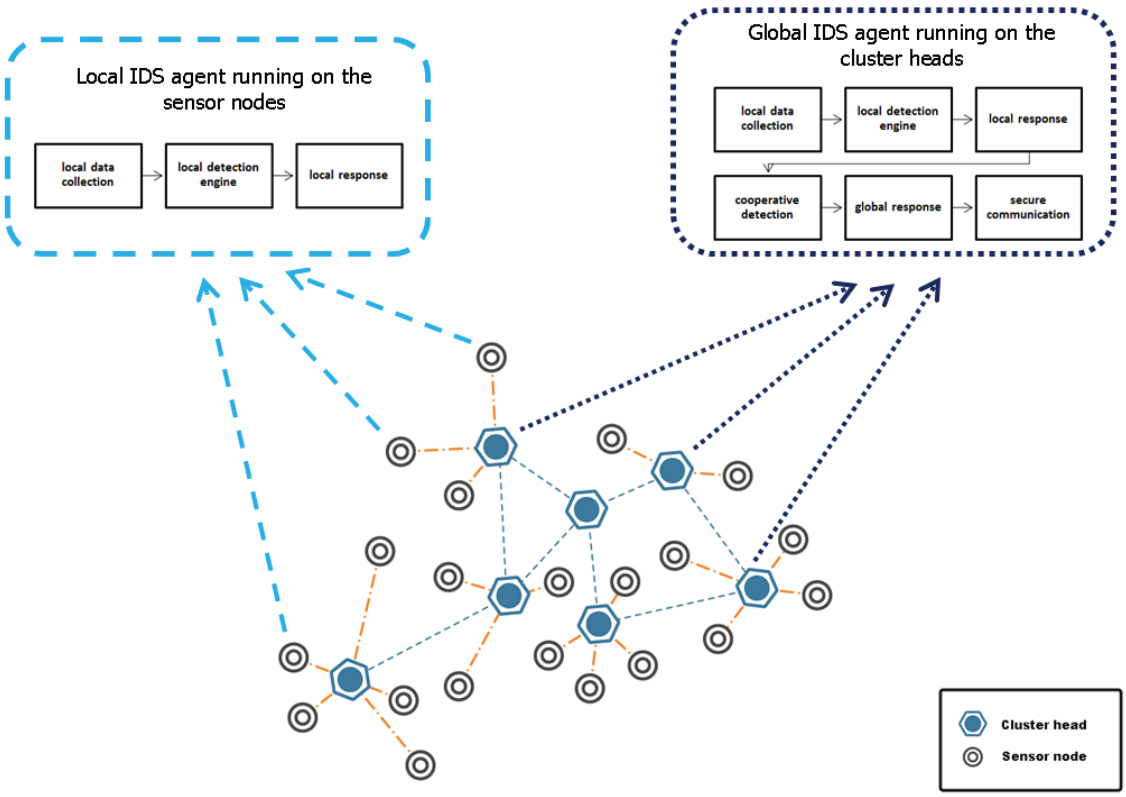

Figure 4.4 Application of an IDS devised for a MANET to a WSN by using clustering approach.

4.4 IDSs proposed for WSNs

Intrusion detection in WSNs is becoming a key research topic addressed in the literature. Therefore, in this section, the research done so far in this field is summarized. Before starting, in Section 4.4.1, the unique challenges of WSNs that make it difficult to apply traditional (designed for wired or generic wireless networks) IDSs are presented. WSNs are special version of MANETs, with very specific design restrictions. Therefore, in Section 4.4.2, the key differences of both networks will be mentioned. Finally, in Section 4.4.3, the state-of-the-art IDSs in the literature of WSNs will be provided. Following all the reviews, we will discuss about advantages and disadvantages of each scheme by providing them in a comparable chart.

4.4.1 Constraints and Research Challenges in WSNs

The proliferation of WSNs led researchers to develop strategies about providing stable communications and networking for distributed network environments, and also about how to secure these

strategies with limited resources. The lack of fixed infrastructure (i.e., gateways, routers, base stations, etc.) makes the design of security related models and algorithms for WSNs more difficult. Bandwidth, throughput, battery life are the scarce resources that need to be used with great consideration. Following is a brief list of constraints and the corresponding challenges they bring to WSNs:

- There is no infrastructure in WSNs to support operations such as communications, routing, real time traffic analysis, encryption, etc.
- Nodes are prone to physical capture, tampering or hijacking which compromises network operations.
- Compromised nodes may provide misleading routing information to the rest of the WSN leaving the network un-operational (blackhole, wormhole, sinkhole attacks).
- Wireless communication is susceptible to eavesdropping, which would reveal important data to adversaries and/or to jamming/interfering, which would cause DoS in the WSN.
- There is no trusted authority; decisions have to be concluded in a collaborative manner.

In designing an IDS for WSNs, these constraints and challenges should be considered.

4.4.2 Differences between MANETs and WSNs

Roman *et al.* [110], highlighted the fact that the IDSs that are designed for MANETs cannot be applied to WSNs directly. Since MANETs are mobile and IDSs for them are designed in the same manner, they will be less effective in a stationary network such as WSNs. Following are basic distinctive features that differentiate WSNs from MANETs:

- Mobility: Compared to mobile MANET nodes, WSN nodes are generally stationary.
- Computational capacity: WSN nodes have limited computational power compared to the MANET nodes. A typical sensor node such as MICAz [111] runs an Atmel ATmega128L processor with a maximum speed of 16 MHz [112], whereas a typical MANET node, such as generic commercial laptop, may have a processor with a maximum speed of 4 GHz [113].
- Communications range: The range of communication is around 20-30 meters for WSN nodes (for MICAz [111]), whereas it is up to 100 meters for MANET nodes (for XBee WiFi module [114]).

- Communications bandwidth: The communication bandwidth is limited to 250 kbps (for a typical MICAz mote [111]) data rate in WSNs, whereas it goes up to 65 Mbps (for a typical XBee WiFi module [114]) data rate in MANETs.

- Lifetime of the power source: WSN nodes have a very limited power source, such as 2 AA sized batteries for MICAz motes [111] (with an approximate energy capacity of 10 Wh), whereas MANET nodes generally have a bigger battery, such as laptop batteries (with an approximate energy capacity of 150 Wh). Obviously, this would affect their lifetime directly. Assuming that their power consumption rates are same, MANETs would have approximately 15 times more life time compared to WSNs.

- Autonomy: In MANETs, every node is managed by a human user, whereas in WSNs every node is autonomous in a sense that it receives and sends data from/to the base station (BS). That BS is generally managed by a human but not the sensor nodes.

- Node density: Node density in WSNs is higher than that in MANETs. On the other hand, WSNs nodes are more susceptible to hardware failures (battery constraints, lacking physical security, etc.), which would decrease the node density with advancing time.

Before adapting an IDS that is designed for a MANET to a WSN, these distinctive features should be considered.

4.4.3 Proposed Schemes

Clustering (Hierarchical) based IDSs: In [9], a hierarchical framework for intrusion detection as well as data processing is proposed. Throughout the experiments on the proposed framework, they highlighted the significance of one-hop clustering. The authors believed that their hierarchical framework was useful for securing industrial applications of WSNs with regard to two lines of defense.

In [10], the authors proposed an isolation table to detect intrusions in hierarchical WSNs in an energy efficient way. Their proposal required two-levels of clustering. According to their experiment, their isolation table intrusion detection method could detect attacks effectively. The problem with this proposal is as follows: The authors claim that each level monitors the other level and report any anomalies to the base station. Since it is a hierarchical network, any alert generated by the lower level nodes must pass through the higher level nodes. In the case that the higher level node

is the intruder, it will not allow the BS to be aware of its misbehavior by simply blocking the alert messages it receives from the lower level nodes.

In [14], an IDS based on clustering approach was proposed. Their proposal also ensured the security of the CHs. In their approach, members of a cluster monitor their CH in a time scheduled manner. In this way, energy for all cluster members is saved. On the contrary, cluster members are monitored by the CH, not by the contribution of cluster members. This also saves the energies of the cluster members. Through simulations, the authors showed that their proposed algorithm is much more efficient compared to other algorithms in the literature. The problem with this approach is its key management mechanism. It's a part of the IDS and helps the IDS to establish pairwise keys among the nodes. The IDS uses these keys through the authentication of the messages. The key management assumes that the nodes are stationary (non-mobile) and the new nodes cannot be added after the pairwise keys are established. This constitutes a handicap for the model considering the fact that WSN may periodically require deployment of the new nodes.

In [115], the authors incorporated a hierarchical IDS model in which the network is divided into clusters and for each cluster, a CH is elected. They issued centralized routing, meaning that every packet of transmitted data will be forwarded to the CH and then to the base station. Their proposal included a method to place intrusion detectors in the CHs so that the entire network is covered with a minimum number of detectors. The authors did not provide any simulation results or any real experimental data. So, it is not clear whether the system would perform as promised.

In [11], a distributed cluster based anomaly detection algorithm was proposed. They minimized the communication overhead by clustering the sensor measurements and merging clusters before sending a description of the clusters to the other nodes. The authors implemented their proposed model in a real-world project. They demonstrated that their scheme achieves comparable accuracy when compared to centralized schemes with a significant reduction in communication overhead.

Distributed and Collaborative IDSs: Krontiris *et al.* [116] proposed a distributed IDS for WSNs based on collaborative neighborhood watching. In a simulation environment, the authors evaluated the effectiveness of their IDS scheme against blackhole and selective forwarding attacks.

In [117], a solution to the problem of cooperative intrusion detection in WSNs was proposed, where the nodes were equipped with local detector modules and have to identify the intruder in a distributed way. The detector modules triggered suspicions about an intrusion in the sensor's

neighborhood. The authors presented necessary and sufficient conditions for successfully exposing the attacker and a corresponding algorithm that is shown to work under a general threat model.

In [81], the proposed IDS used a specification based detection algorithm. The authors used a decentralized approach of detection in which intrusion detectors were distributed among the network (their distance was one-hop, covering the entire network). The collected information and its processing were performed in a distributed fashion. They claimed that this distributed approach was more scalable and robust compared to a centralized approached owing to the fact that the intrusion detectors had different views of the network by being distributed to all over the network.

Statistical Detection based IDSs: Ngai *et al.* [118] presented an algorithm to detect the intruder in a sinkhole attack. The proposed algorithm first finds a list of suspected nodes and then effectively identifies the intruder in the list through a network flow graph. The algorithm implements a multivariate technique (statistical - parametric technique) based on the chi-square test. Effectiveness and accuracy of the proposed algorithm is verified by both numerical analysis and simulations. The authors claimed that their algorithm's communication and computational overheads are reasonable for WSNs.

In the proposed algorithm of [119], the sensor network adapts to the norm of the dynamics in its natural surroundings so that any unusual activities can be singled out. In order to achieve this, they employ a hidden Markov model. The authors claimed that their proposed algorithm is easy to employ, requiring minimal processing and data storage. The functionality and practicality of the algorithm is shown through experimental scenarios. The proposed algorithm sifts out any unusual readings by using the statistical approach. So it is a very specific kind of IDS that is mainly focused on the accuracy of the data gathered rather than the security of the nodes or the links.

In [120], the authors proposed a real time, node based anomaly detection algorithm that observes the arrival processes experienced by a sensor node. They developed an arrival model for the traffic that can be received by a sensor node and devised a scheme to detect anomalous changes in that arrival process. The detection algorithm kept short term statistics using a multi-level sliding window event storage scheme. In this way the algorithm could compare arrival processes at different time scales. The authors claimed that their algorithm was resource aware and has low complexity.

Game Theory based IDS in WSNs: In [12] and [13], Agah *et al.* considered attack and detection as both participants of the game and formulated strategies for both parties. In order to increase detection probability, strategies were normalized into a non-cooperative, non-zero game model. Both

schemes focused on determining the weakest node in the network and then providing strategies to defend that node. The problem with this approach was that there might be multiple intrusions to the WSN and only one of them would be caught by the IDS while leaving others undetected.

Anomaly Detection based IDSs: In [121], Rajasegarar *et al.* provided a survey article about the state of the art in anomaly detection techniques for WSNs. They suggested for the researchers (for anomaly detection) to consider the inherent limitations of WSNs in their design so that the energy consumption in sensor nodes is minimized and the lifetime of the network is maximized. In [122], the same authors proposed a solution to the problem of minimizing the communication overhead in the network while performing in-network computation when detecting anomalies. Their approach to this problem is based on a formulation that uses distributed one-class quarter-sphere support vector machines to identify anomalous measurements in the data. Data vectors are mapped from the input space to a higher-dimensional space for further investigations. The authors implemented their proposal in a real-world project and they claimed that their model was energy efficient in terms of communication overhead while achieving comparable accuracy to a centralized scheme.

Bhuse and Gupta [123] proposed lightweight methods to detect anomaly intrusions in WSNs. Their main idea was to re-use the already available system information (such as neighbor lists, routing tables, sleep/wake-up schedules, receive signal strength indication, MAC layer transmission schedules) that was generated at various OSI layers of a network protocol stack, especially the physical, MAC and routing layers. In order to have a better detection rate, the authors proposed multiple detectors monitoring different layers of the OSI stack. This is not feasible for WSNs, because intrusion monitoring in different layers and sustaining the coordination of these monitors may rapidly deplete the scarce resources of the WSN. Besides, the authors proposed their schemes for outsider attacks only, ruling out the insider attacks. This is inadequate choice, because sensor nodes in a WSN are very vulnerable to insider attacks such as physical capture attack, Sybil attack, etc.

Onat and Miri [124] provided an IDS for WSNs that was based on detection of packet level receive power anomalies. The detection scheme was focused on transceiver behaviors and packet arrival rates of the neighboring nodes of a particular node. WSNs are rarely mobile and therefore they have a stable communication pattern when compared to MANETs. The authors exploited this specific distinction. Each node built a simple statistical model of its neighbors' behavior and used

this statistics to detect any abnormal changes in the future. The proposed model worked well to detect impersonation attacks.

Watchdog based IDS: Roman *et al.* [110] provided guidelines about application of IDSs (that are designed for MANETs) to static WSNs. Then they propose an IDS for WSNs called 'spontaneous watchdogs', in which the neighbors are optimally monitored and where some nodes choose to independently monitor the communications in their neighborhood.

Reputation (Trust) based IDS: Wang *et al.* [125] proposed an IDS for WSNs that uses packet marking and then heuristic ranking algorithms to identify most likely bad nodes in the network. Each packet is encrypted and padded so as to hide the source of the packet. The packet mark is added in each packet such that the data sink can recover the source of the packet and then figure out the dropping ratio associated with every sensor node. According to their simulations, most of the bad nodes could be identified by their *heuristic ranking algorithm* with small false positive rate.

Bao *et al.* [126] proposed a hierarchical trust management for WSNs to detect selfish and malicious nodes. Authors developed a probability model utilizing *stochastic Petri nets technique* to analyze the protocol performance and validated subjective trust against objective trust obtained based on ground truth node status. Their trust-based IDS algorithm outperforms anomaly-based IDS algorithms in the detection probability percentage while maintaining sufficiently low false positive rates.

4.4.4 Issues and Comments Concerning the Proposed Schemes

IDSs proposed for WSNs are summarized in Table 4.2 including their required network architecture, detection technique and highlighting features of each scheme. Accordingly, the following conclusions are drawn for the proposed IDSs in WSNs:

- In hierarchical, clustering based IDSs, clustering algorithms may consume considerable amount of the network's energy through the formation of the clusters. After the clusters are formed and the CHs are elected, CHs may constitute a single point of failure and they have to be secured. Besides, if the CH is not a special node (more powerful), then the overhead of being a CH will diminish its resources very quickly.

- Agent based IDSs reduce the network load and latency. On the other hand, they cause high energy consumption of the nodes they are working on. Communication cost between

Table 4.2 Comparison of the IDSs proposed for WSNs.

Proposed system	**Architecture**	**Detection technique**	**Highlighting features**
Da Silva *et al.* [81]	Distributed	Rule based approach (interval rule)	Scalable, robust and fast intrusion detection.
Roman *et al.*, [110]	Distributed and Cooperative	Spontaneous watchdogs	Relies on the broadcast nature of sensor communications and takes advantage of the high density of sensors being deployed in the field.
Chen *et al.* [10]	Hierarchical	Rule based approach	Uses monitoring group of nodes and routing tables for detection
Su *et al.* [14]	Hierarchical	Rule based approach (packet dropping rate)	Saves energy, extends the network lifetime. On the other hand, new nodes cannot be added to the network.
Strikos [115]	Hierarchical	Rule based approach	Combined already existing approaches, in order to achieve a more complete solution. Neither simulation results, nor real world experimental results are provided.
Rajasegarar *et al.* [11]	Hierarchical	Specification based approach, data clustering (standard deviation from the average inter-cluster distance)	Achieved comparable performance with the centralized schemes.
Krontiris *et al.* [116]	Distributed and Cooperative	Rule based approach (packet dropping rate)	Detects only blackhole and selective forwarding attacks. Besides, proposed solution works only when there is one attacker.
Krontiris *et al.* [117]	Distributed and Cooperative	Specification based approach	Proposed solution works only when there is one attacker.
Ngai *et al.* [118]	Centralized (BS)	Statistical based anomaly detection (parametric), routing pattern anomalies	Specified to detect Sinkhole attacks only.
Doumit and Agrawal [119]	Hierarchical	Statistical anomaly based approach (parametric), hidden Markov model	Focused on the accuracy of the data gathered, rather than the security of the nodes or the links.
Onat and Miri [120]	Stand alone	Statistical based anomaly detection (real time traffic on the nodes, arrival process)	Keeps short term dynamic statistics using a multi-level sliding window event storage scheme. The scheme works on each node, therefore the detections are local and nodes are not aware of the attacks globally (network-wide).
Agah *et al.* [12, 13]	Hierarchical	Game theory along with Markov decision process	Only one of the clusters of the network is monitored at a time. This leaves the rest of the network unprotected.
Bhuse and Gupta [123]	Stand-alone	Rule based approaches (for physical, MAC, routing and application layers)	Proposed lightweight techniques that would detect anomalies at all layers of a network stack in WSNs.
Onat and Miri [124]	Distributed and Cooperative	Statistical anomaly based approach (average receive power and average packet arrival rate)	Exploits the stability of the neighborhood information of the WSN nodes.
Rajasegarar *et al.* [122]	Distributed	Anomaly based approach, support vector machine	Minimizes communication overhead while performing in-network anomaly detection.
Wang *et al.* [125]	Centralized (data sink)	Reputation based approach	Uses heuristic ranking algorithms to identify most likely bad nodes in the network.
Bao *et al.* [126]	Hierarchical	Reputation based approach	Uses high scalable cluster-based hierarchical trust management protocol to effectively identifying the selfish and malicious nodes.

agents and coordinator, or in between agents, may cause congestion and bottle neck in the network.

- Rule based IDSs are simple to install and easy to operate. On the other hand, they need continuous rule updates in order to cope with the new released attacks.
- Data mining based IDSs can detect unknown attacks. Unfortunately they have high computational complexity and high energy consumption requiring large amounts of data samples. Besides, they also need efficient analytic tools to analyze mass audit data and a mass storage.
- In game theory based IDSs, the detection rate can be adjusted by the network security administrator through changing the parameters. The problem with this system is that it is non-adaptive and requires human intervention for a stable operation.

4.5 Future Directions in the Selection of IDS for WSNs

Energy consumption of the IDSs is an important issue from a system design point of view. WSNs consume energy through sensing the surrounding phenomena, processing the sensed information and transmitting the resultant data. Therefore, the IDSs need to spend the least amount of energy as possible to spare enough energy for the crucial operations of the WSN. As a result of this low energy consumption requirement of WSNs, it is beneficial to use a hierarchical model for IDSs. This means that the network would be divided into clusters, each of which will have a CH. Accordingly, the energy consumption will be minimized by avoiding the need for all the nodes to send data to the BS. Besides, high energy consuming IDS algorithms would run only on the CHs which would save energy on the rest of the nodes and ultimately increase the total lifetime of the network.

Since there are a variety of intrusion detection algorithms available, the selection of the intrusion detection technique would be specific to the requirements of the intended application; i.e, the attacks that need to be detected, the accuracy of the detection (percentage of the false positives and true positives), and the duration of the detection time.

Our suggestion for the selection of the IDS for WSNs will be application specific (various suggestions for different applications):

- For the mobile applications, where sensor nodes are in movement, we recommend the usage of distributed and cooperative IDS schemes, as they are scalable, robust and fast.

Da Silva *et al.*'s [81], Roman *et al.*'s [110] and finally Onat and Miri's [124] proposed schemes are recommended as the most promising ones among those presented in Table 4.2.

- For the stationary applications, where there is a centralized computing unit at BS or at data sink, we recommend the usage of centralized IDS schemes, as they are powerful and can detect whole range of attacks. Among the schemes presented in Table 4.2, Wang *et al.*'s [125] proposed scheme is recommended for adopting or can be a good starting point to build on it.

- For the cluster based applications, where the network is divided into clusters, the usage of hierarchical IDS schemes is suggested. Among the schemes presented in Table 4.2, Su *et al.*'s [14] work is recommended, if the network is stable and no nodes are to be added. Otherwise, Bao *et al.*'s [126] work is suggested, as it is efficient for the scalable and dynamic network topologies.

For the researchers that are considering to simulate and compare the performances of the various IDS schemes, Adaobi *et al.*'s work [127] would be a good starting point. In their work, authors provide a case scenario on how to simulate an attack against a WSN and evaluate the performance of an anomaly-based IDS. Authors simulate their scenario in ns-2 simulation environment [65], with AODV protocol. They provide 4 metrics (namely, true positives, true negatives, false positives, and false negatives) calculated by analyzing the packet delivery ratio while changing the pulse rate.

To the best of our knowledge, there is no paper published regarding the effects of the IDSs on the energy consumption of WSNs. For the researchers that are considering to evaluate the cost of the IDS schemes on the WSNs, this would be a good topic to research.

4.6 Conclusions

In this chapter; IDSs along with their classifications, design specifications and requirements, are briefly introduced. Secondly, IDSs that are proposed for MANETs are presented and their applicability to WSNs, are discussed. Thirdly, IDSs proposed for WSNs are discussed and their distinctive features are highlighted in a comparable chart, followed by the comments regarding IDSs that would be applicable to WSNs are presented. Finally, in order to help researchers in the selection

of IDS for WSNs, recommendations of promising proposed schemes are provided along with future directions for this research.

CHAPTER 5 :

POWER AND CONNECTIVITY AWARE CLUSTERING FOR WIRELESS SENSOR NETWORKS

5.1 Introduction

As mentioned in earlier chapters, WSNs are characterized by severely constrained computational and energy resources, and an ad hoc network operational environment. They pose unique challenges, due to limited power supplies, low transmission bandwidth, small memory sizes and limited energy; therefore, networking techniques used in traditional networks cannot be adopted directly [24]. So, new ideas and approaches (algorithms) are needed in order to increase the overall performance of the network, especially in terms of total life-time. **Clustering**, is one of those techniques that is very useful to WSNs in data aggregation, and is the main focus of this chapter.

A clustered-WSN is typically as shown in Figure 5.1. Each cluster is a group of interconnected sensor nodes with a dedicated node called cluster head (CH). CHs are responsible for the management of the cluster such as scheduling of the medium access, dissemination of the control messages, and the most importantly, data aggregation [23]. The size of a cluster is defined as the hop distance from the CH to the farthest node in the cluster. For example, in a 3-hop cluster, the distance between the CH and the farthest node is 3-hops (4 nodes are in the path including the end points). The clustered network shown in Figure 5.1 has a 1-hop distance in between CHs and the member sensor nodes.

Clustering is the process of grouping the nodes in a network that are within a specified hop distance or have some shared common properties into clusters and electing CHs for each cluster. This election can be made permanent (static clustering) or repeated in some certain time intervals (dynamic clustering). Clustering is used in many applications of wireless sensor networks in order to reduce the traffic load on the nodes through data aggregation process, to prolong total network life-time, to balance the data traffic in the network and finally to increase the scalability (allows the deployment of hundreds or thousands of nodes). Besides, clustering helps us to increase security of

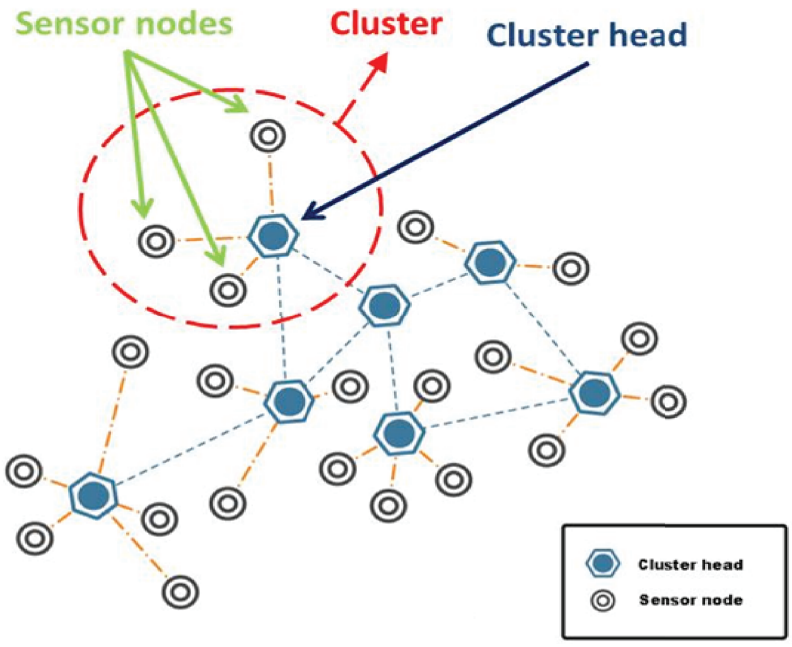

Figure 5.1 A typical clustered WSN.

the network by allowing implementation of complex cryptography algorithms. By using clustered networking approach, power consuming algorithms (such as data aggregation) would be run on the CHs and this would help us to significantly improve the total life-time of the network.

In this chapter, we investigate clustering algorithms that are proposed for WSNs and propose a new clustering algorithm that is both power and connectivity aware. The rest of the chapter is organized as follows: Section 5.2 provides a description of the related work available in the literature. Section 5.3 presents Kachirski *et al.*'s connectivity aware clustering algorithm and Section 5.4 provides the revised and improved version of that algorithm. Our proposed power and connectivity aware clustering algorithm is presented in Section 5.5. Section 5.6 provides the comparison of both schemes and also presents the details of our simulation environment. In Section 5.7, we discuss the observations regarding the effects of the clustering on the performance of the WSNs. Finally, Section 5.8 concludes the chapter and outlines future work.

5.2 Related Work

There are plenty of clustering algorithms available in the literature that are proposed for wireless networks. In this section, we present the most widely used clustering algorithms and mention their advantages and disadvantages:

Low Energy Adaptive Clustering Hierarchy (LEACH) [17], is a distributed clustering algorithm in which nodes make autonomous decisions without any centralized control. Cluster formation is cyclically performed and history information of the previous CHs are stored. CHs are assigned as a result of a random procedure, where each node can declare itself as a CH with some probability. Energy levels of the nodes are included as a factor in the CH selection whilst connectivity of the nodes are ignored. Therefore, it is not guaranteed that every node is within K-hops of a CH. This is the main concern of LEACH, which may cause some nodes to be segregated from the rest of the network during the time period in between the two election cycles. Another drawback of LEACH is due to the assumptions that not only the network size and the number of CHs are known in advance but also all nodes are very well synchronized (in order to ensure that CHs can be re-elected periodically to balance the energy consumption). These are very specific assumptions that might not fit well to the real life applications of WSNs.

In [18], Bandyopadhyay *et al.* propose a distributed and randomized clustering algorithm similar to the LEACH. The proposed algorithm also aimed at energy efficiency and its difference from the LEACH is that it provides hierarchical (multi-level) clustering as well. Other than that, the proposed algorithm holds the same concerns and the drawbacks as LEACH does.

In [21], Jia *et al.* present an energy consumption balanced clustering algorithm (LEACH-EP) for WSNs that is based upon LEACH algorithm. It introduces energy factor in CH electing threshold, and optimizes the election probability of CH. As in the case of LEACH, LEACH-EP comes with specific assumptions as well.

Energy Efficient Clustering Scheme (EECS) [20] is also based upon LEACH algorithm and aims at energy efficiency. Its difference from the LEACH is the set-up phase of the clusters (cluster formation). The proposed algorithm holds the same concerns and the drawbacks as LEACH does.

In Hybrid, Energy-Efficient, Distributed Clustering (HEED) [19] approach; CHs are periodically selected according to a hybrid of their residual energy and a secondary parameter, such as a

node's proximity to its neighbors or node degree. HEED does not make any assumptions about the distribution or the density of the nodes, nor their connectivities.

Evenly Distributed Clustering (EDC) algorithm [16] distributes clusters uniformly and minimizes the number of clusters. It considers the connectivity of the nodes with the K-hop parameter. It is a heuristic approach, in which each node only exchanges its head selection with its neighbors. Based on neighbors' selection results, each node chooses the nearest head as its CH. The drawback of this algorithm is that it does not consider the density of the nodes in a network. In order to increase the life-time of the network, it is important to elect more CHs in the dense areas of the network. However, the algorithm is aimed at distributing the cluster heads evenly to the network deployment field.

In [23], Brust *et al.* present algorithms for cluster head candidate selection that are based on topology (location) of the nodes. The algorithms aim to avoid selecting nodes located close to the network partition border because those nodes are more likely to move out of the partition, thus cause a clusterhead re-election. By using the connectivity information, they propose three algorithms to find the strong, weak, bridge and board nodes in the network. Authors do not provide any information on how to select the CHs among their selection of nodes (strong, weak, bridge and board nodes). Overall, this classification of nodes for CH selection would be useful for the mobile ad hoc networks (MANETs) where mobility is the prime factor that changes the network topology. However, the network topology in WSNs is quite stable compared to MANETs, and therefore this kind of node classification is unnecessary for CH selection.

Energy Efficient Unequal Clustering (EEUC) [22] is proposed for periodically data gathering WSNs. It partitions the nodes into clusters of unequal size, and clusters closer to the base station have smaller sizes than those farther away from the base station. This way, CHs closer to the base station can preserve some energy for inter-cluster data forwarding.

Hierarchical clustering proposed in [9] is a framework based on two-level clustering; multi-hop clusters for data aggregation (the first level clustering) and 1-hop clusters for intrusion detection (the second level clustering). Although the idea sounds promising in some applications of WSNs (especially the industrial applications); the details of the formation algorithms for the multi-level clustering were missing (we assume that this was left as a future work).

Kachirski *et al.*'s [15] clustering algorithm is based on the connectivity of the nodes in the network. The higher connectivity (neighbors) a node has, the higher probability of it to be elected as the CH

of a certain neighborhood (cluster). This algorithm is one of the best choice for us to work on for several reasons: First of all, it did not require probabilistic approach on clustering and therefore the result of the clustering would cover the whole network. Secondly, the connectivity of the nodes are the main concern on the election of CHs, which is reasonable. In general the nodes that have more connections would be rather elected as CHs. Finally, the algorithm is easily implementable, which allows the proof of the theoretical work on both hardware and simulation environment.

The only missing part in Kachirski *et al.*'s [15] clustering algorithm was the power awareness. Therefore, in this article, we propose our clustering algorithm that is built upon the revised version of Kachirski *et al.*'s algorithm. Our algorithm is both power and connectivity aware, that is why, it provides maximum throughput while saving energies of the nodes, therefore significantly increases the life-time of the network.

5.3 Kachirski *et al.*'s Connectivity based Approach for Clustering

Kachirski *et al.*'s [15] clustering algorithm is based on the connectivity of the nodes in the network. The higher connectivity (neighbors) a node has, the higher probability of it to be elected as the cluster head (CH) of a certain neighborhood (cluster).

In order to demonstrate the principles of the algorithm, consider the network shown in Figure 5.2. Here we assume that each node has $1-hop$[1] connectivity, meaning that each node can communicate with its direct neighbors that are in $1-hop$ communications distance (in terms of radio range). In order to elect the CHs, these are the steps to be followed:

1. Let C_i denote the number of established connections (nodes that are one-hop away in our case) for node i, with total number of N nodes in the network. Each node calculates its own C_i value (as shown in Figure 5.3, note that the numbers written each node represents total number of neighbors for each node) and sends it to all its neighbors.

2. After receiving C_k values from its neighbors k (where $k \neq i$, for all $i = 1 \ldots N$), a node i calculates the connectivity index (S_i) as shown in Equation 5.1:

$$S_i = C_i + \sum_k C_k \tag{5.1}$$

[1]The same method would be applied in the case of multiple-hop (2,3,..., etc.) connections if needed.

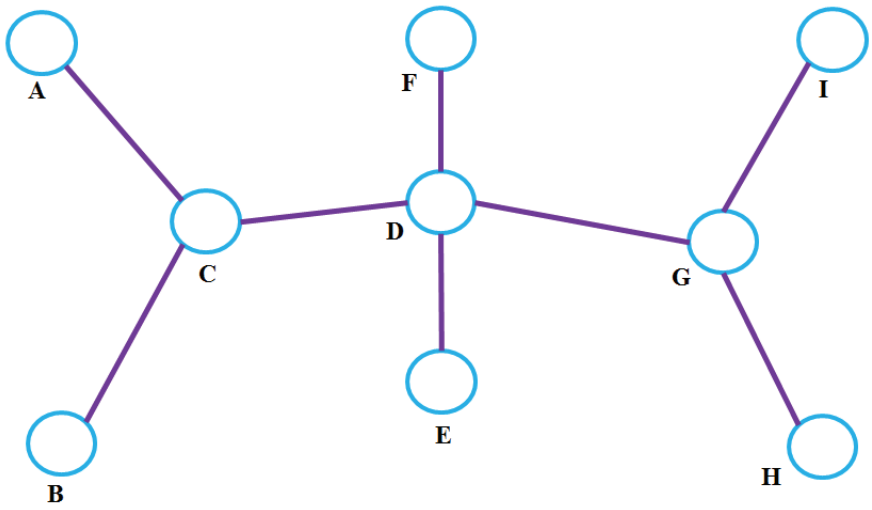

Figure 5.2 A typical 9-node WSN.

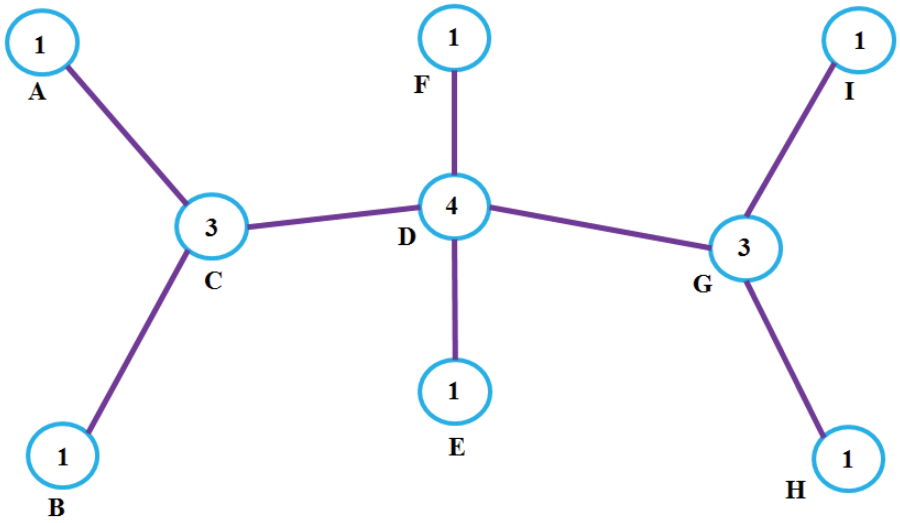

Figure 5.3 Established connections graph, indicating total number of one-hop neighbors for the WSN shown in Figure 5.2.

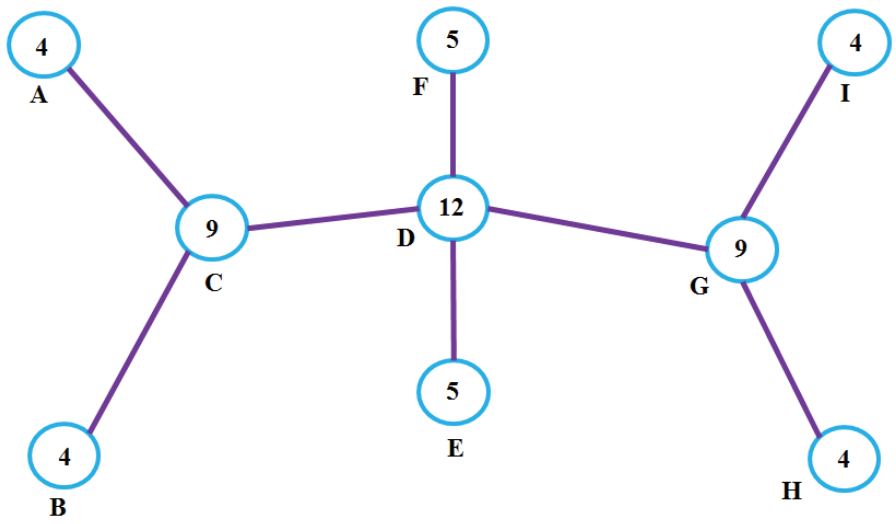

Figure 5.4 Connectivity index graph (1-hop) of the WSN shown in Figure 5.2.

Each node calculates its own connectivity index according to Equation 5.1. For the network shown in Figure 5.3, the connectivity indices would be as shown in Figure 5.4.

3. Each node broadcasts its connectivity index (S_i) to all other nodes with a time to live (TTL) value equivalent to time spent through one hop communication.

4. Each node then has to participate in a voting session in which the cluster head will be determined. Each node votes for the node that has the highest S_i value, as a result of the broadcast operation in Step-3.

5. After the voting procedure, if a node receives at least one vote, it is assigned as the cluster head. After the voting session, the network members in Figure 5.4 select their cluster heads as shown in the Figure 5.5[2].

5.4 Revised Version of Kachirski *et al.*'s Connectivity based Approach for Clustering

In the specific case of the network shown in Figure 5.2, there are nine members of the network and three members (out of nine) are elected as cluster heads, as a result of the voting procedure (see Figure 5.5). As the network connectivity increases, we expect to have more connected members in the network resulting in less number of selected cluster heads. As an example, for the same configuration of the network in Figure 5.2, if we use $2-hop$ connectivity for the node communications, we obtain

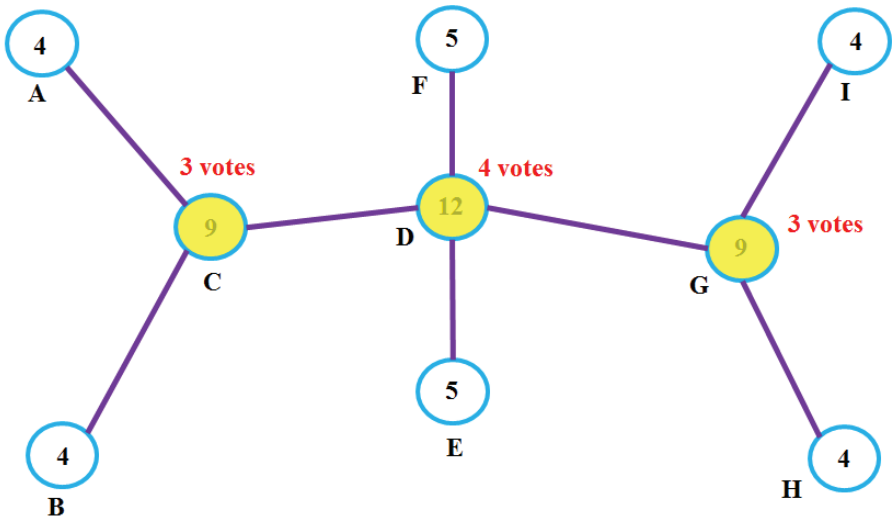

Figure 5.5 Elected cluster heads (1-hop) (shown in yellow color) and associated number of votes, after the voting session for the WSN shown in Figure 5.2.

the neighborhood graph as shown in Figure 5.6. By applying Equation 5.1 and then performing the voting session, the connectivity index graph (denoted on the nodes) and the cluster head selections would be as shown in Figure 5.7[2].

This is quite an interesting result, since we were expecting to have less cluster heads by increasing the connectivity (number of maximum hops). This happens because of a fault in the voting procedure of Kachirski *et al.*'s [15] clustering algorithm: We realized that throughout the voting procedure, nodes are not voting for themselves and this may result in more cluster heads to be elected than needed. In order to fix this problem, we revised Kachirski *et al.*'s clustering algorithm by letting the nodes voting for themselves (if they have the highest connectivity index).

We applied the revised scheme to our example network (see Figure 5.2) and the result of the voting scheme is shown in Figure 5.8[3]. As a result, the total number of cluster heads is one, resulting in less cluster heads (instead of three) as we expected.

[2]Cluster heads are highlighted with yellow color and also the votes they received are noted on top of them in red color writing.

[3]Cluster heads are highlighted with yellow color and also the votes they received are noted on top of them in red color writing.

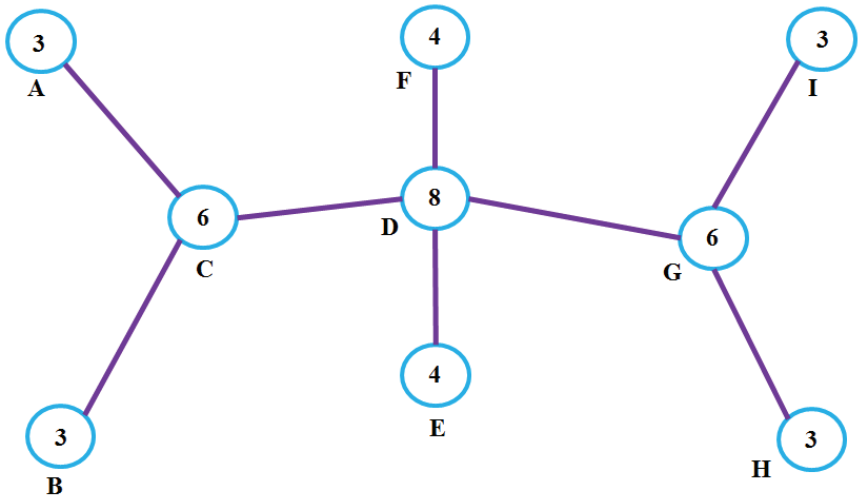

Figure 5.6 Established connections graph, indicating total number of two-hop neighbors for the WSN shown in Figure 5.2.

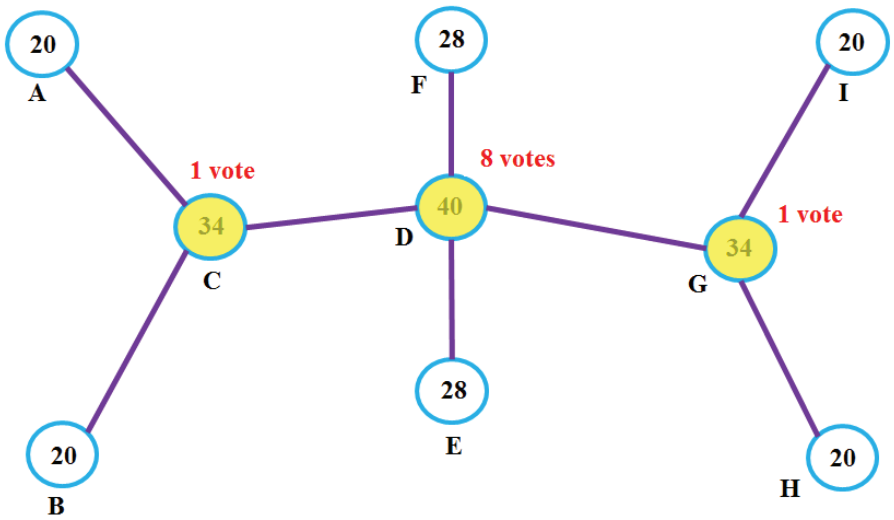

Figure 5.7 Connectivity index graph and elected cluster heads (2-hop) of the WSN shown in Figure 5.2.

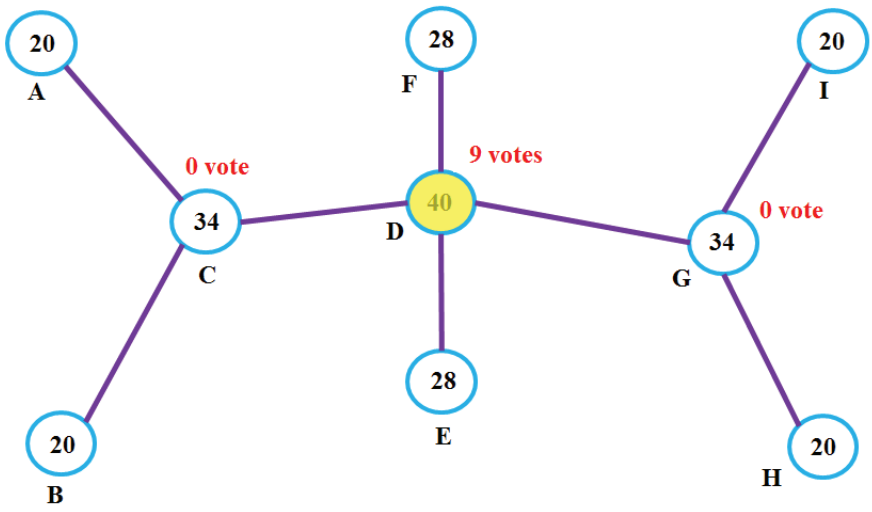

Figure 5.8 Elected cluster heads (2-hop) of the WSN shown in Figure 5.2 by using the Kachirski *et al.*'s revised clustering scheme.

5.5 Our Power and Connectivity Aware Approach for Clustering

In WSNs, energy is one of the scarce resources that needs to be conserved. As a result of the clustering algorithms, elected cluster heads become the highest energy consuming nodes of the network, since they perform operations related to data aggregation, security, routing, etc., on behalf of the other nodes.

Kachirski *et al.*'s [15] clustering algorithm (see Section 5.3) and its revised version (see Section 5.4) does only consider a node's connectivity with its neighbors while determining a cluster head. But it does not consider any parameter regarding the energies of the nodes.

In order to increase the total life-time of a WSN, energy (power) levels of the nodes also should be considered while determining the cluster heads. Therefore, we propose a power and connectivity aware clustering algorithm based upon Kachirski *et al.*'s [15] clustering algorithm. We achieve this by introducing power level readings through connectivity index calculations (step-2). Our scheme determines the cluster heads according to this calculations. Voting scheme follows the revised version of the Kachirski *et al.*'s clustering algorithm (node may vote for themselves).

The description of our proposed scheme is as follows:

1. Let C_i denote the number of established connections for node i, with total number of N nodes in the network. Each node calculates its own C_i value and sends it to all its neighbors.

2. After receiving C_k values from its neighbors k (where $k \neq i$, for all $i = 1 \ldots N$), a node i calculates the connectivity index (S_i) as shown in Equation 5.2:

$$S_i = C_i + \sum_k C_k + \beta \times P_i \tag{5.2}$$

   With the help of Equation 5.2, each node's connectivity index not only carries information regarding its connectivity with its neighbors but also informs the power level of that particular node[4].

   Consider the network shown in Figure 5.2. The connectivity indices and the voting results were shown in Figure 5.8. Here, we re-calculate the connectivity indices for that network according to Equation 5.2 as shown in Figure 5.9. Here, each green writing over the nodes represents the power level (percentage) of that node at the time that the clustering calculation is done.

3. Each node broadcasts its connectivity index (S_i) to all other nodes with a time to live (TTL) value equivalent to time spent through one hop communication.

4. Each node then has to participate in a voting session in which the cluster head will be determined. Each node votes for the node that has the highest S_i value (nodes are allowed to vote for themselves), as a result of the broadcast operation in Step-4.

5. After the voting procedure, if a node receives at least one vote, it is assigned as the cluster head. After the voting session, the network members in Figure 5.9 select their cluster heads as shown in the Figure 5.10[5].

When a WSN uses our power and connectivity aware clustering approach, we expect two parameters to effect the total-life time of the network:

[4] In our calculations, P_i value represent the battery level of each particular node, i.e., 1.00 means the battery level of the node is 100% of its maximum level.

[5] Cluster heads are highlighted with yellow color and also the votes they received are noted on top of them in red color writing.

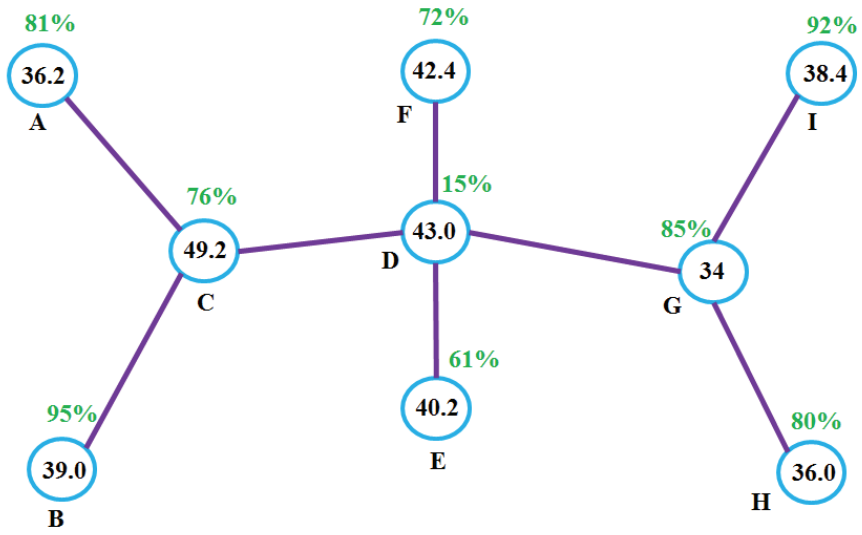

Figure 5.9 Connectivity index graph (2-hop) of the WSN shown in Figure 5.2, as a result of our power and connectivity aware clustering approach.

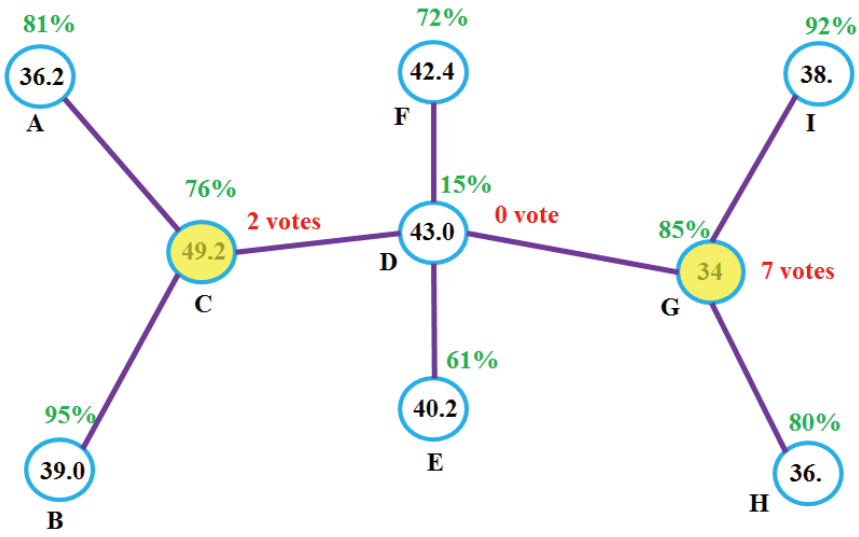

Figure 5.10 Elected cluster heads (2-hop) of the WSN shown in Figure 5.9 by using our power and connectivity aware clustering scheme.

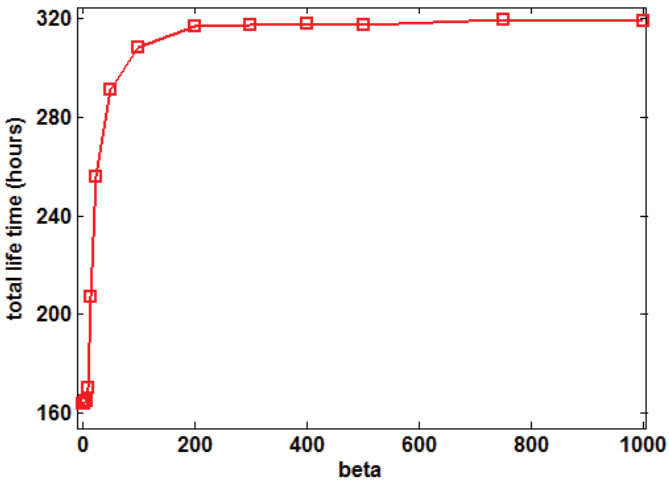

Figure 5.11 Total life-time vs. beta, for the WSN shown in Figure 5.9 by using our power and connectivity aware clustering scheme.

- *Power factor (β):* An optimum value of β can be determined by fixing every parameter in the network and then by observing the total life time of the network with the change of β. Here, it is important to note that, β is not correlated to the connectivity term (C_k) in Equation 5.2 (i.e., power level of a node is not directly related to the total number of established connections to a node but to the throughput measured on those links).

- *Period of clustering (τ):* It is the time period that determines the renewal of the cluster heads by re-applying the clustering algorithm. An optimum value of τ can be determined by fixing every parameter in the network and then by observing the total life time of the network with the change of τ.

As an example, we simulated our power and connectivity aware clustering algorithm on the network shown in Figure 5.9 with the simulator discussed in the next section (Section 5.6). Figure 5.11 shows the behavior of the total life time with the change of β; whereas Figure 5.12 shows the behavior of the total life time with the change of τ. According to the result of the simulations, we may conclude that, for the network configuration of Figure 5.9 and the parameter selection shown in Section 5.6.2; the optimum value of β is 200 (the total life-time curve in Figure 5.11 saturates

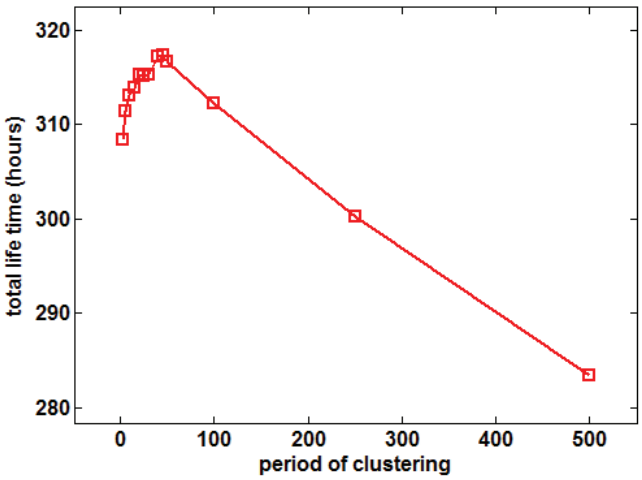

Figure 5.12 Total life-time vs. period of clustering, for the WSN shown in Figure 5.9 by using our power and connectivity aware clustering scheme.

for $\beta \geq 200$) and the optimum value of τ is 45 (the total life-time curve in Figure 5.12 gets the maximum value at $\tau = 45$ and starts decreasing as τ becomes bigger or smaller than this value).

5.5.1 Applicability of Our Power and Connectivity Aware Clustering Algorithm to Nowadays WSNs

Our power and connectivity aware clustering algorithm is very applicable to nowadays WSNs. Because, current Commercial Off-The-Shelf (COTS) nodes, such as Wasp motes [35], provide the power reading of it's batteries (as a percentage) as an available information which could be sent to other nodes. This information would be used directly by our power and connectivity aware clustering algorithm in order to determine the cluster heads.

5.6 Comparison of Both Schemes in Terms of Total Life-time of the Wireless Sensor Network

In order to evaluate and compare the effect of both Kachirski *et al.*'s (revised) and our power and connectivity aware clustering algorithms on the total life time of the WSNs, we created a simulation environment in MATLAB. The details of the simulation environment are as follows:

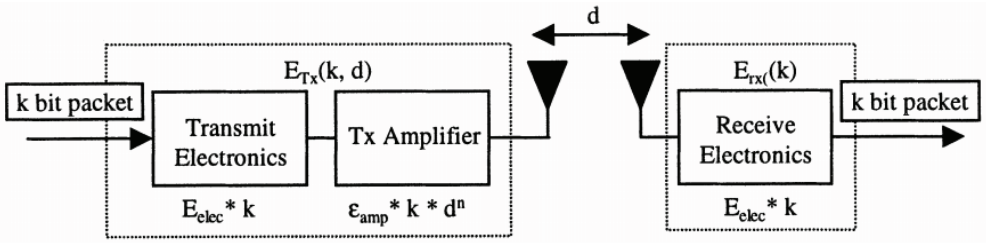

Figure 5.13 Radio energy dissipation model used in our simulations [128].

5.6.1 Energy Consumption Calculations

For energy consumption calculations, we followed Heizelman *et al.*'s work [128]. We assume a simplified model (since radio wave propagation is mostly non-stable and difficult to model) for the radio hardware energy dissipation, where the transmitter dissipates energy by running the radio electronics and the power amplifier, whereas the receiver dissipates energy by running the radio electronics only, as shown in Figure 5.13.

We consider two different channel models depending on the distance between the transmitter and the receiver:

1. *Near Field (free space - fs) Channel Model:* If the distance between the transmitter and the receiver is less than a threshold (d_0) then this model is used (also called d^2 power-loss model).

2. *Far field (multipath - mp) Channel Model:* If the distance between the transmitter and the receiver is greater than a threshold (d_0) then this model is used (also called d^4 power-loss model).

According to [21], the threshold value for the distance is calculated as follows:

$$d_0 = \frac{\sqrt{\epsilon_{fs}}}{\sqrt{\epsilon_{mp}}} \tag{5.3}$$

where ϵ_{fs} and ϵ_{mp} are constants related to free space loss and multipath loss, respectively.

In order to transmit m-bit data to a distance of d, the radio spends:

$$E_{Tx}(m, d) = E_{Tx-elec}(m) + E_{Tx-amp}(m, d) = \begin{cases} mE_{elec} + m\epsilon_{fs}d^2, & d < d_0. \\ mE_{elec} + m\epsilon_{mp}d^4, & d \geq d_0. \end{cases} \tag{5.4}$$

In order to receive the same m-bit data, the radio spends:

$$E_{Rx}(m) = E_{Rx-elec}(m) = mE_{elec}. \tag{5.5}$$

The energy spent on the radio electronics circuitry, E_{elec}, is due to the digital modulation (transmitter-side), digital demodulation (receiver-side), error correction codes and filtering; whereas the amplifier energy, E_{Tx-amp} is due to the electromagnetic spreading of the signal into the air and depends on the distance as mentioned above (see Equation 5.3.).

Let's assume that each cluster head has N member nodes. Cluster head dissipates energy by receiving the data from member nodes, aggregating those data , and finally transmitting the aggregate data to the BS. We assume that BS is located far away from the nodes and therefore transmission between the cluster head and the BS follows the far field channel model (d^4 power-loss model). During a single data frame, we calculate the energy dissipated in the cluster head as follows:

$$\begin{aligned} E_{CH} &= \{E_{aggregating\_data\_from\_member\_nodes}\} + \{E_{transmit\_aggregate\_data\_to\_BS}\}. \\ &= \{N(mE_{elec} + mE_{DA})\} + \{mE_{elec} + m\epsilon_{mp}(d_{toBS})^4\}. \end{aligned} \tag{5.6}$$

where m represents total number bits in a data frame, mE_{DA} represents the energy dissipated during aggregating m-bit data and finally d_{toBS} represents the distance between the cluster head and the BS.

Assume that each member node is located in the near field of the cluster head, so that near field channel model (d^2 power-loss model) will be used for calculating the energy dissipated during data transmission from the member node towards the cluster head. Therefore, we calculate the energy dissipated in each member node as follows:

$$E_{member\_node} = mE_{elec} + m\epsilon_{fs}(d_{toCH})^2. \tag{5.7}$$

where d_{toCH} represents the distance between the member node and the cluster head and therefore it takes different values for each node.

Table 5.1 Values for the energy consumption related parameters used through our simulations.

Parameter	Value
E_{elec}	$50nJ/bit$
ϵ_{fs}	$10pJ/bit/m^2$
ϵ_{mp}	$0.0013pJ/bit/m^4$
d_0	$87.7m$
E_{DA}	$5nJ/bit/data$

For all our simulations in the text, we followed [128] and used the values for the energy consumption related parameters as shown in Table 5.1.

5.6.2 Simulation Parameters

Here are the parameters that we used during the simulation:

- Each node in the network is identical to each other and has a starting energy of 2 Joules.
- There is a base station (BS) located outside of the network to collect the data from cluster heads.
- The deployment area is $100m$ x $100m$.
- Data flow from nodes to cluster heads. Cluster heads aggregate the data and then forward to the BS.
- The header size for each frame is 200 bits.
- The header size for each frame is 4000 bits.
- Data rate is 1 frame per 10 minutes (0.1 frames/min).
- We consider a packet drop rate of 5% for the transmission of each data frame due to the collisions and multi-path fading.
- We consider a stationary network, meaning that both BS and the nodes are not moving.
- Since we will be comparing two clustering schemes, we ignored the cost associated with the formation (voting and etc.) of the clusters.
- Each simulation is run 1000 times and an average value of the life time (that falls into 95% confidence interval) is calculated. For example, for the simulation result of Figure 5.19

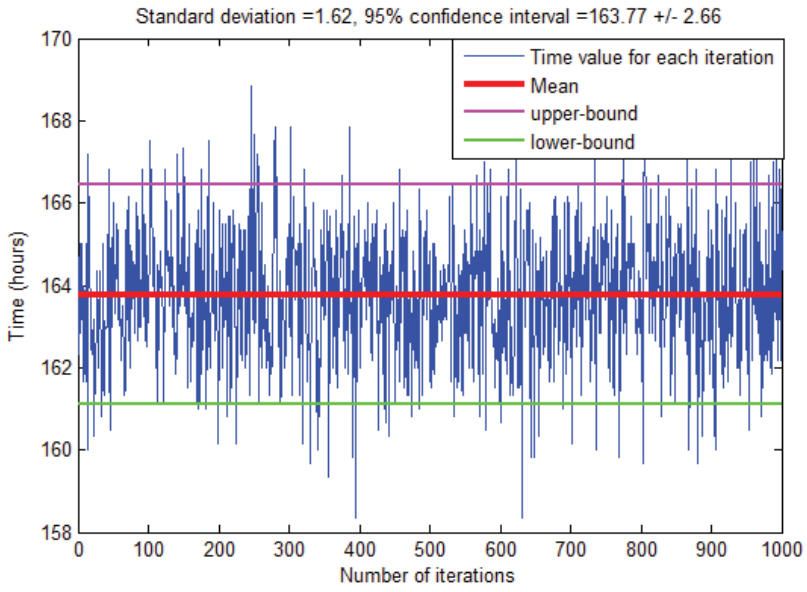

Figure 5.14 Distribution plot of the simulation time.

(Section 5.6.4), the distribution of the total life-time is shown in Figure 5.14. Here, the lower bound and upper-bounds of the confidence interval (95%) , as well as the mean value are shown on the graph. Figure 5.15 and Figure 5.16 show the corresponding histogram plot and quantile-quantile plot, respectively. In these figures, the first plot represents our simulation data, and the second plots represent the normal distribution (Gaussian) that has the same mean value as of our data. From these figures, we observe that our simulation result shows a normal distribution (in quantile-quantile plot, our data cumulates on the x=y line). Therefore, we calculated the mean value for each simulation in this text to represent all the values resulted in 1000 iterations.

5.6.3 Coordinates

Throughout our simulations, we assumed that both BS and the nodes are stationary, therefore their coordinates are fixed. For the following sections, coordinates of the nodes and the BS will be as shown in Figure 5.17. Here, circular shapes represent the nodes (blue ones are the member nodes and the red ones are the cluster heads) whereas the square shape represents the BS. The red lines represent the connection between the cluster heads and the BS, whereas the blue lines represent the connections between the cluster heads and their member nodes. The whole deployment area is

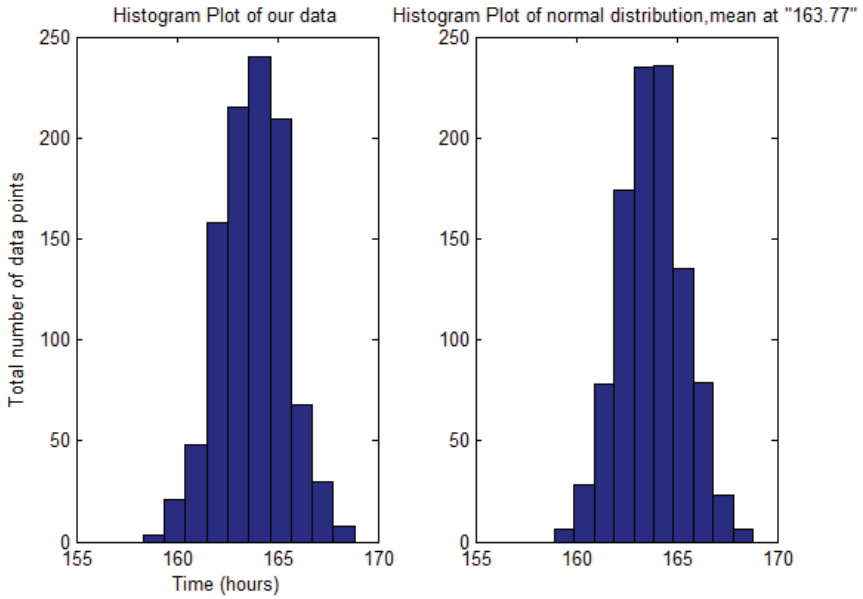

Figure 5.15 Histogram plot of the simulation time compared to the normal distribution.

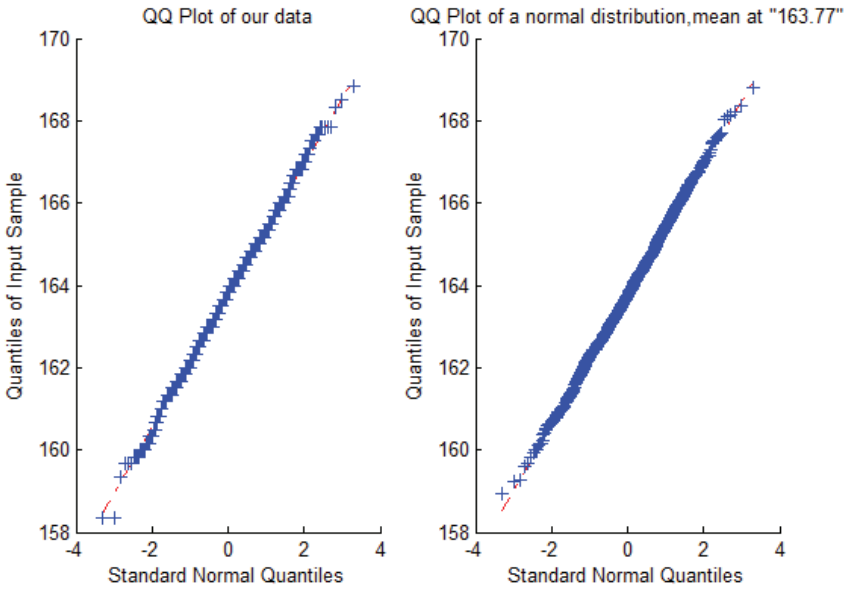

Figure 5.16 Quantile-Quantile plot of the simulation time compared to the normal distribution.

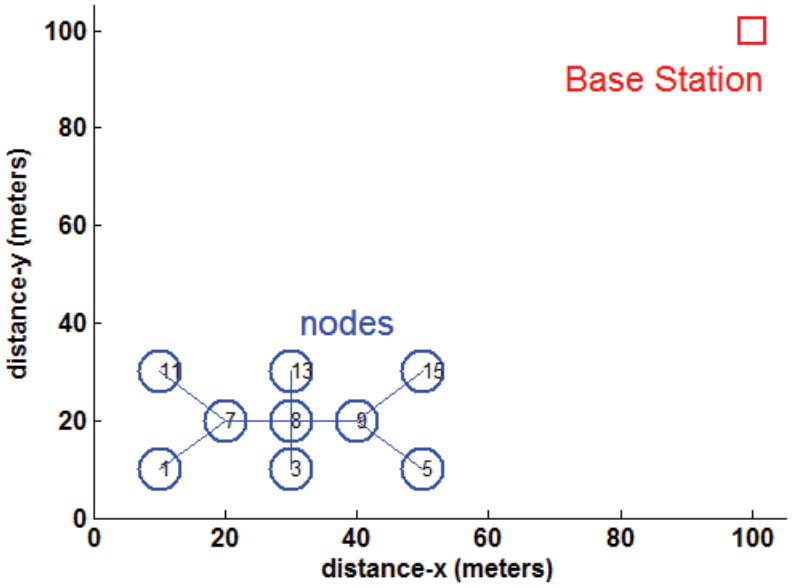

Figure 5.17 Plot of coordinates of the nodes and the BS throughout the simulations.

$100m$ x $100m$ and the location of the BS is $[100m, 100m]$. The nodes are deployed to the area with the following boundaries: $[10m, 10m]$, $[10m, 30m]$, $[50m, 10m]$, $[50m, 30m]$.

5.6.4 Energy Consumption of Kachirski *et al.*'s Clustering Algorithm (revised version)

We ran Kachirski *et al.*'s clustering algorithm (revised version) on our simulator with the parameters shown in Section 5.6.2 and the coordinates shown in Section 5.6.3. We consider 1-hop connectivity for all nodes in the network. Figure 5.18 shows the total number of neighbors for each node (including the connection paths), connectivity indices, results of the voting along with the elected cluster heads.

Figure 5.19 shows the energy consumption performance of the mentioned algorithm with respect to time. We stopped the simulation, whenever a single node dies (runs out of battery power), and we call this time as the "total-life time of the network", since at this point the network starts to disintegrate (segregation starts).

In Figure 5.19, we can see that node-8 depleted its energy faster than other nodes and therefore determined the network's life-time as 163.77 hours.

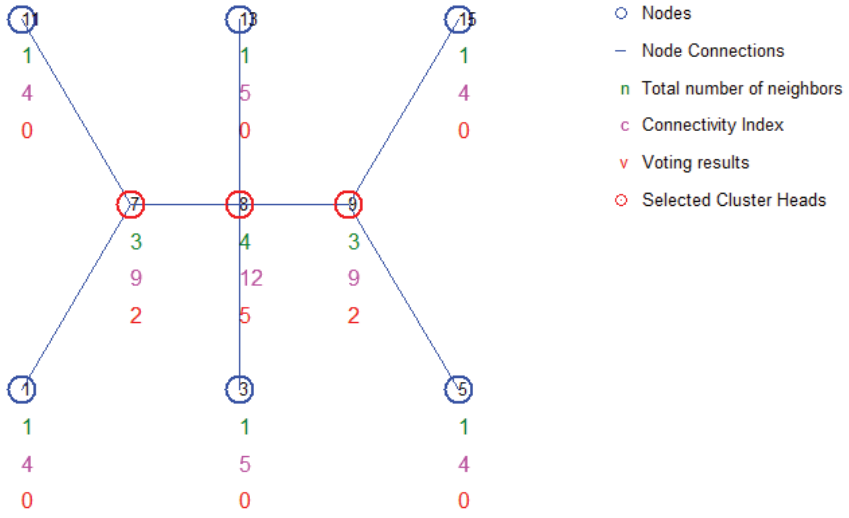

Figure 5.18 Cluster head selection of a 9-node WSN with Kachirski *et al.*'s algorithm (revised version) for 1-hop connectivity case.

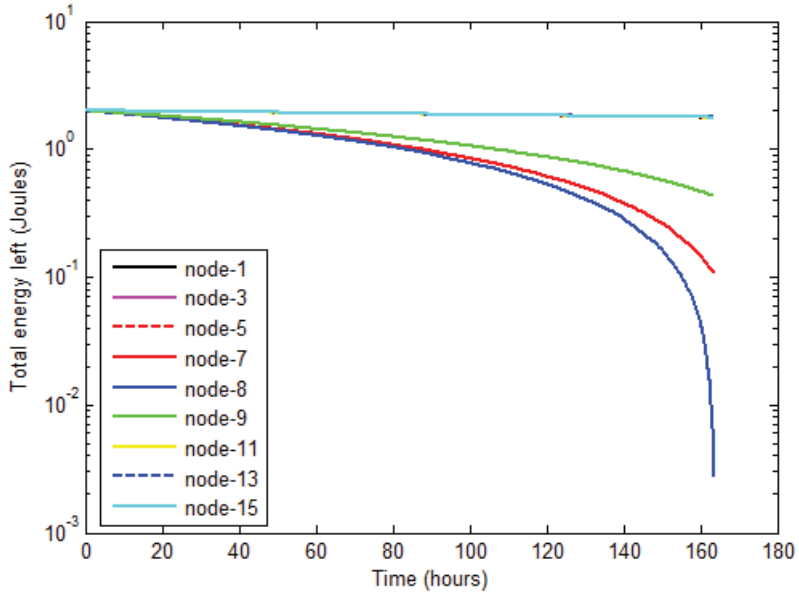

Figure 5.19 Energy consumption graph of Kachirski *et al.*'s clustering algorithm (revised version) for 1-hop connectivity case.

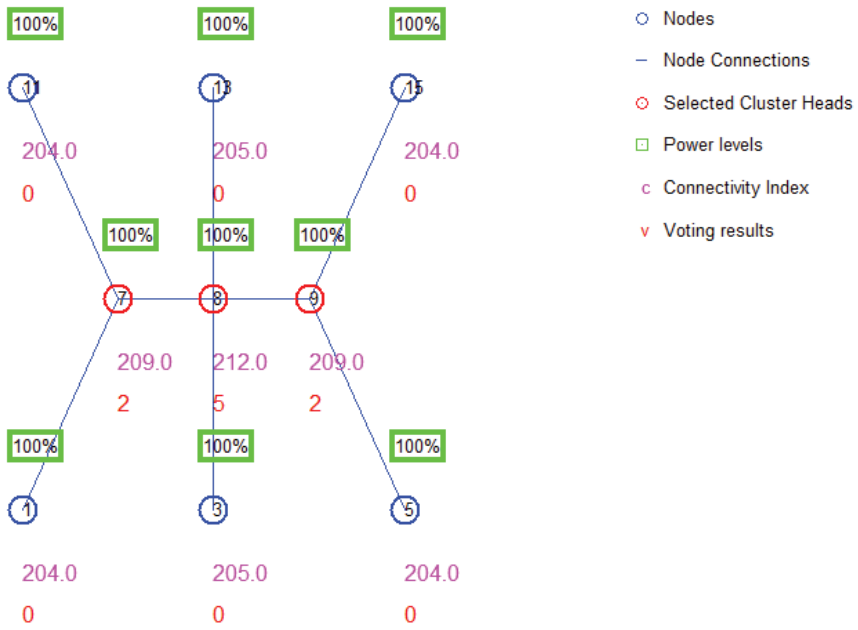

Figure 5.20 Cluster head selection of a 9-node WSN with our algorithm at time $t = 0$.

5.6.5 Energy Consumption of Our Power and Connectivity Aware Clustering Algorithm

We ran our power and connectivity aware clustering algorithm (revised version) on our simulator with the parameters shown in Section 5.6.2 and the coordinates shown in Section 5.6.3. We consider 1-hop connectivity for all nodes in the network. As mentioned in Section 5.5, we selected β as 200 and τ is 45, in order to achieve the maximum life-time. Figures 5.20 and 5.21 show the result of the clustering algorithm at times $t = 0$ and $t = t_1(t_1 > 0)$, respectively. Figure 5.22 shows the energy consumption performance of our algorithm with respect to time. In Figure 5.22, we can see that node-8 depleted its energy faster than other nodes and therefore determined the network's life-time as 316.66 hours.

In order to provide further proof of performance improvement on life time, we repeated the same simulation setup with different network topologies with 7 nodes, 9 nodes and finally 15 nodes (see Figure 5.23). We ran both clustering algorithms on these networks in 3 different maximum number of hops: 1, 2 and 3. The resulting relative performance improvements on the life-time of the network are as shown in Table 5.2. Accordingly we conclude that, as the maximum number of hops increases,

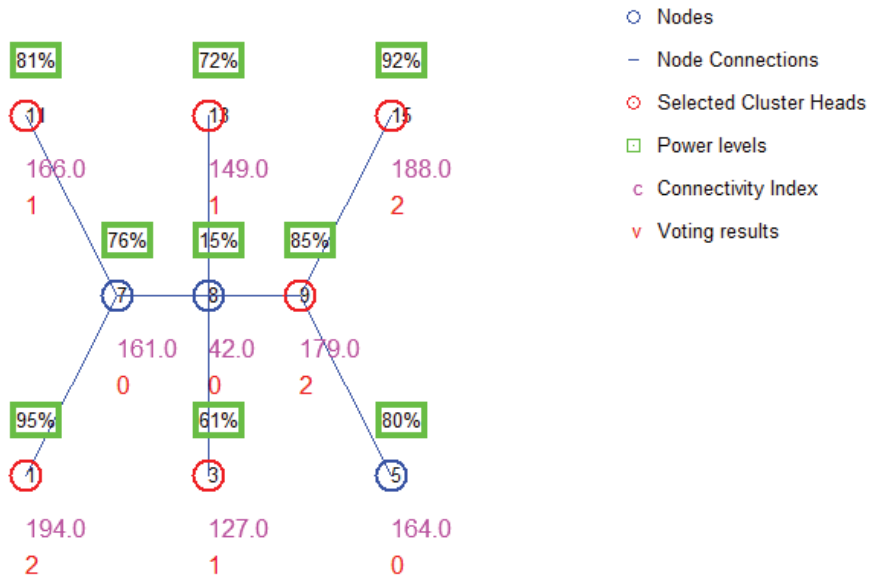

Figure 5.21 Cluster head selection of a 9-node WSN with our algorithm at time $t = t_1(t_1 > 0)$.

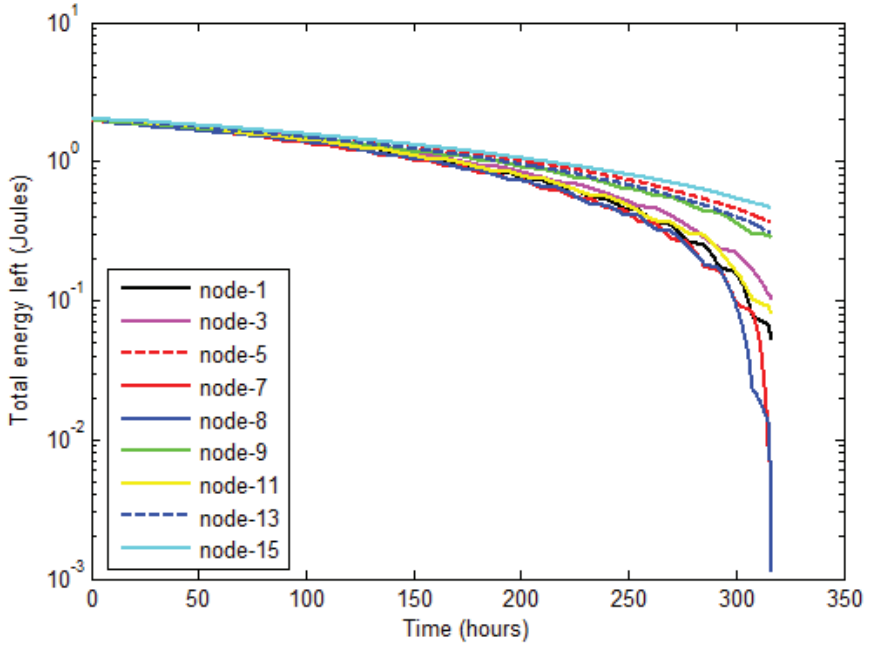

Figure 5.22 Energy consumption graph of our power and connectivity aware clustering algorithm for 1-hop connectivity case.

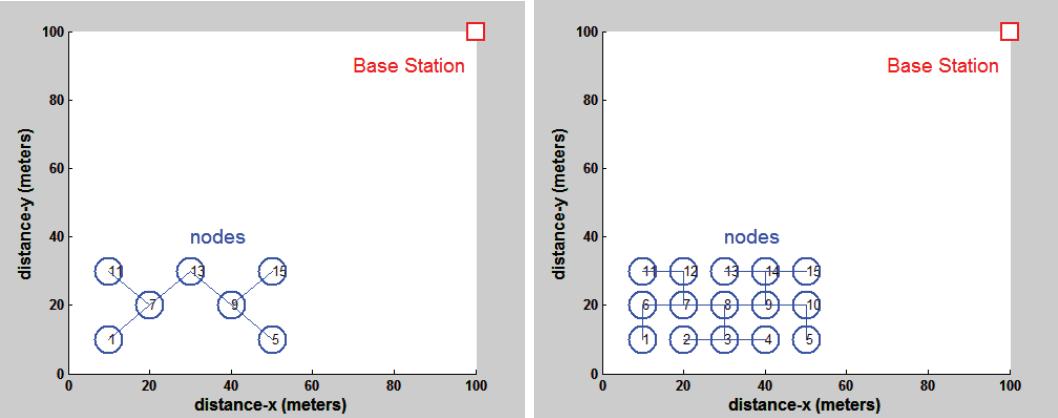

Figure 5.23 Different network topologies with 7 and 15 nodes.

Table 5.2 Relative performance improvements (%) on the life-time of the network when our algorithm is used.

Maximum hops	for 7-node network	for 9-node network	for 15-node network
1 hop	86	93	85
2 hops	234	313	256
3 hops	366	463	438

our clustering algorithm becomes more beneficial. This is because, more nodes become eligible to be elected as CHs as the maximum number of hops increases. According to our simulations, the relative performance improvement of clustering algorithm varies between 85-93% for 1-hop neighborhood, 234-313% for 2-hop neighborhood, and finally 366-463% for 3-hop neighborhood, respectively.

5.7 Some Observations on the Effect of Clustering to the Network Performance

5.7.1 Effect of Maximum Number of Hops on Total Number of Cluster Heads

As the maximum number of hops increases, the nodes in the network achieve more communications with the other member nodes, and as a result the network requires less number of cluster heads. To support this hypothesis, we have run the cluster head selection algorithm on a 15 nodes network for 3 different maximum number of hops: 1,2 and 3. The resulting total number of neighbors, connectivity indices, voting results and the elected cluster heads are shown in Figure 5.24, Figure 5.25 and Figure 5.26, respectively. Accordingly, Figure 5.27 shows the plot of maximum number of hops vs. total number of elected CHs, for a 15 node network.

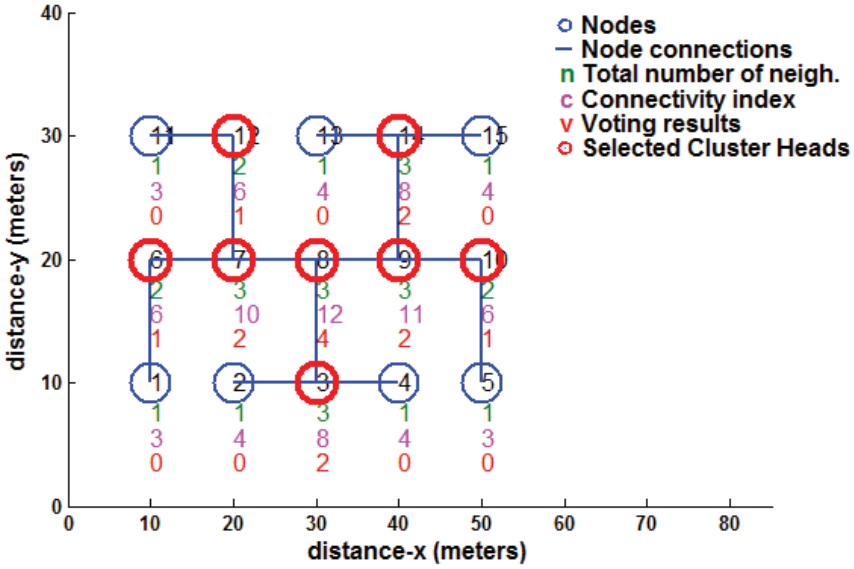

Figure 5.24 Clustering of 15-node network, 1-hop communications case.

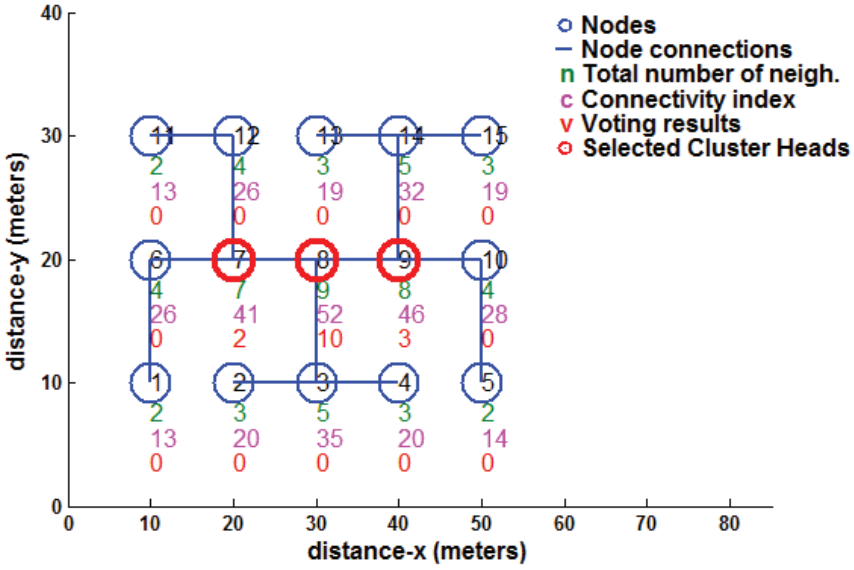

Figure 5.25 Clustering of 15-node network, 2-hop communications case.

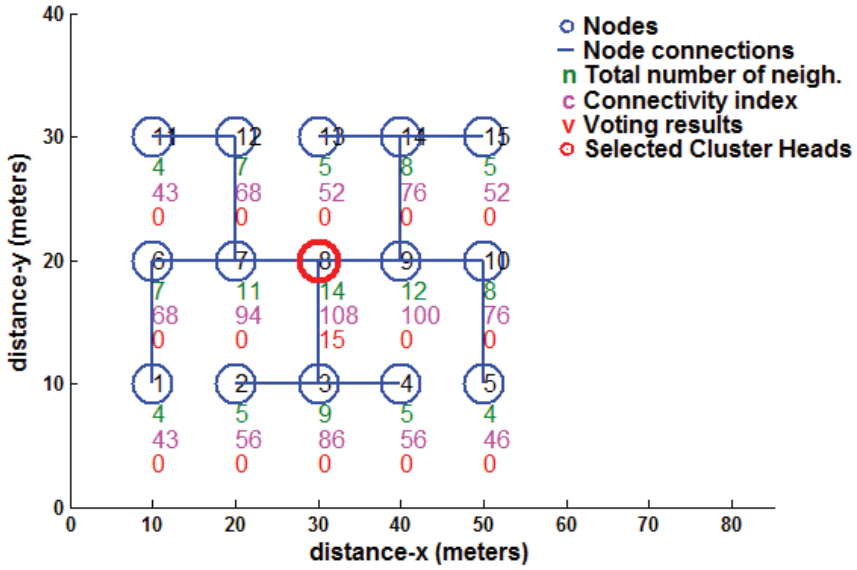

Figure 5.26 Clustering of 15-node network, 3-hop communications case.

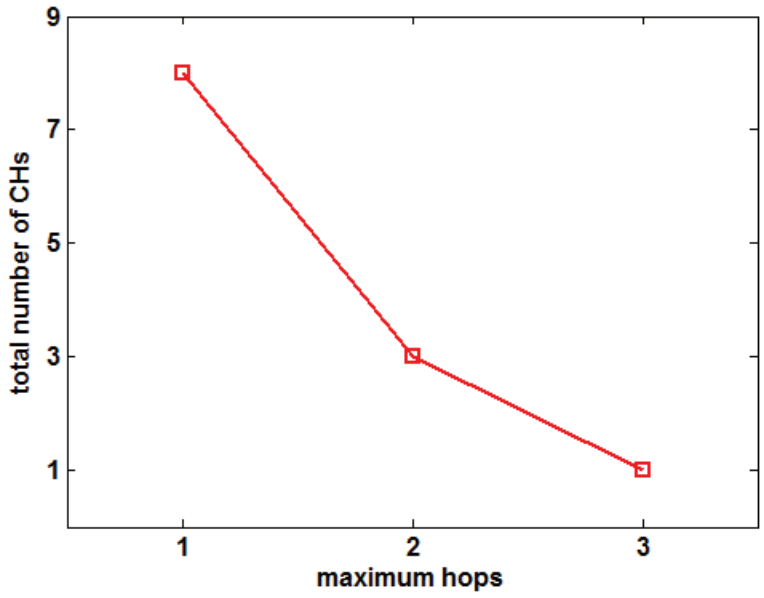

Figure 5.27 Maximum number of hops vs. total number of CHs for a 15 node network.

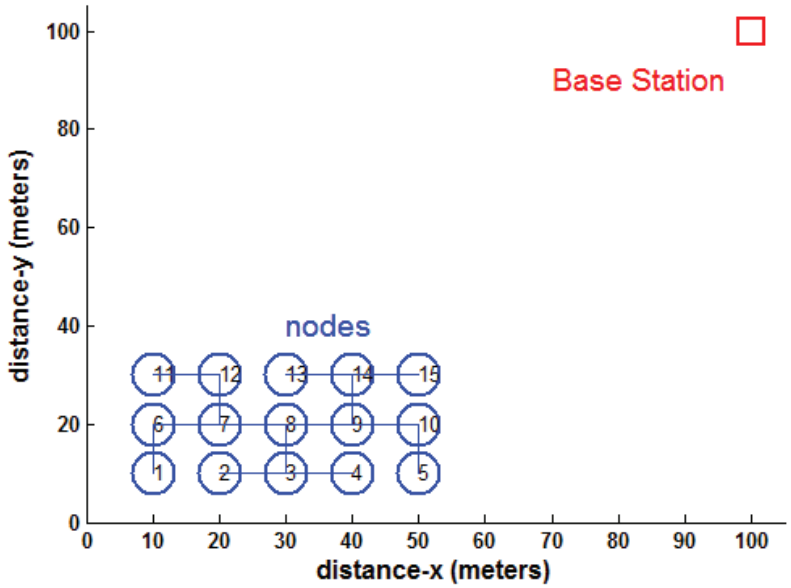

Figure 5.28 Coordinates of the nodes and the BS.

5.7.2 Effect of Total Number of Cluster Heads (maximum hops) on Total Life-time of the Network

The coordinates of the BS and the WSN nodes are as shown in Figure 5.28. BS is located in the far field of the WSN, meaning that the distance between CHs and the BS is greater than 87.7 meters. The simulation parameters are as same as shown in Section 5.6.2. β is chosen as 200 and the τ as 40 frames. By using these parameters and coordinates, we run the simulation for 10 cases of the maximum hops: 0,1,...,8 and 9. Figure 5.29 shows the behavior of total life-time of the network with respect to maximum hops. Accordingly, we conclude that as the maximum hops increases, the total life-time of the network increases. From the slope of the curve, we deduct that this increase saturated at a certain number of hops. This is the point where each node can reach any node in the network (6-hops in this case).

5.7.3 Effect of Total Number of Nodes in a Cluster on Total Life-time of the Network

We wondered about the effect of total number of the nodes in the network on the total life-time of the network. To investigate this, we considered the same network shown in Figure 5.28 along with the simulation parameters same as Section 5.7.2. The only parameter that is different here is the

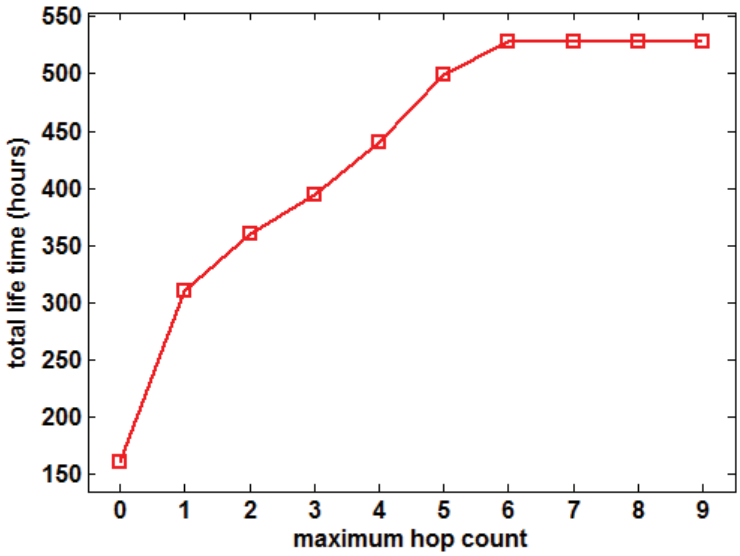

Figure 5.29 Maximum hops vs. total life-time of the network.

maximum hops. We kept it constant and equal to 3. Then we started the simulation with 15 nodes and then each time we removed one of the end nodes, repeated the simulation till we are left with 1 node in the network. As a result, Figure 5.30 shows the behavior of total life-time of the network with respect to the total number of nodes in the network. Accordingly, we conclude that there is a certain number of nodes (6 nodes in our case) in the network that provide the network to achieve maximum total life-time (519.15 hours in our case).

5.7.4 Effect of Data Rate on Total Life-time of the Network

Here, we investigated the effect of frame rate of the nodes in the network on the total life-time of the network. We considered the same network shown in Figure 5.28 along with the simulation parameters same as Section 5.7.2. The only parameter that is different here is the maximum hops. We kept it constant and equal to 3. Then we started the simulation with 15 nodes and then each time we changed the frame rate, repeated the simulation for various frame rates. As a result, Figure 5.31 shows the behavior of total life-time of the network with respect to the frame rate of the nodes in the network. Accordingly, we conclude that as the frame rate increases, total life-time decreases. This is an expected result, because as the frame rate increases, more packets are sent between the nodes thus more energy is consumed.

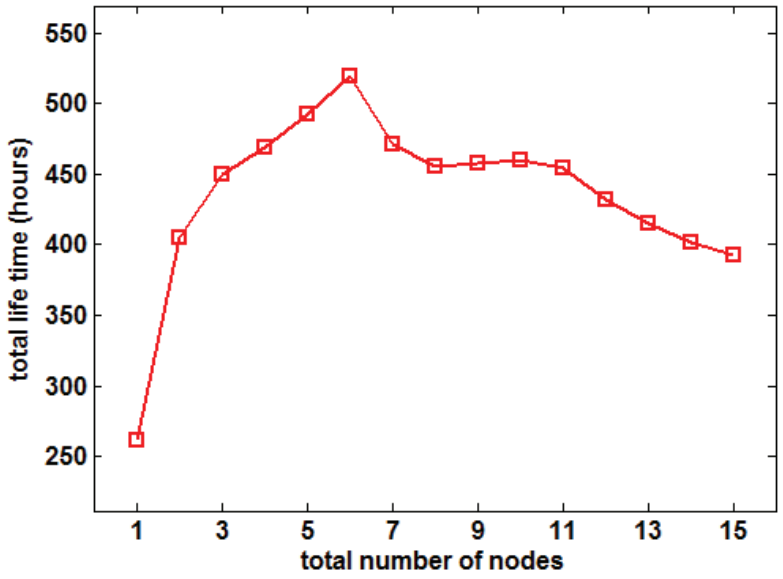

Figure 5.30 Total number of nodes vs. total life-time of the network.

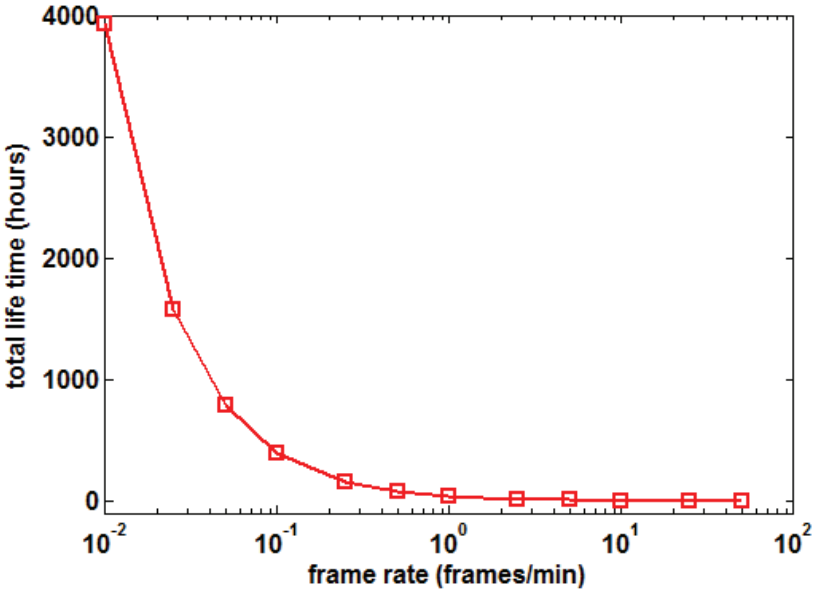

Figure 5.31 Frame rate vs. total life-time of the network.

5.7.5 Conclusions from the Observations

There are 3 major trade-off situations that need to be balanced when implementing solutions (i.e., security, etc.) to a clustered WSN:

1. There is a trade-off between 'maximum hop count' and 'total number of CHs'. As the maximum hop count increases, total number of CHs decreases and vice versa.

2. There is a trade-off between 'total number of CHs' and 'total life-time of the network'. There is an optimum number of CHs which leads network to survive the most life-time possible (without having any partioning/segregation).

3. There is a trade-off between 'data rate (frames/minute)' and 'total life-time of the network'. As the data rate increases, more data need to be processed and more packets need to be transmitted causing more power to be spent, therefore the total life-time of the network decreases.

5.8 Conclusions and Suggestions for Future Research

In this chapter, the energy consumption simulation results of revised version of Kachirski *et al.*'s clustering algorithm and our proposed power and connectivity aware clustering algorithm are provided. According to these results, our proposed power and connectivity aware clustering algorithm out performed revised version of Kachirski *et al.*'s clustering algorithm in terms of energy efficiency and also total life-time of the network.

According to the simulation results, with our proposed power and connectivity aware clustering algorithm, relative performance improvement (compared to the revised version of Kachirski *et al.*'s clustering algorithm) in total life-time of the network varies between 85-93% for 1-hop neighborhood, 234-313% for 2-hop neighborhood, and finally 366-463% for 3-hop neighborhood, respectively.

Here, note that mobility can also be included as an another parameter in cluster head calculations (in Equation 5.2) for MANETs. For example, highly mobile nodes (Wasp motes [35] provide 3-axis accelerometer reading which would be used to measure mobility) maybe elected as cluster heads, because they might be in contact with most of the nodes in a certain amount of time. Since WSNs are mostly stationary, mobility is not considered in the calculations presented in this chapter and left as a future work to be considered.

CHAPTER 6 :

AN INTRUSION DETECTION SYSTEM BASED ON MULTI LEVEL CLUSTERING FOR HIERARCHICAL WIRELESS SENSOR NETWORKS

6.1 Introduction

In this chapter, an Intrusion Detection System (IDS) framework for hierarchical Wireless Sensor Networks (WSNs) that is based on multi-level clustering is proposed. The framework is based upon the clustering algorithm that is proposed in this dissertation (the nodes use our proposed clustering algorithm that is presented in Chapter 5, while forming their clusters). Our proposed IDS framework provides two types of intrusion detection approaches, namely "Downwards-IDS (D-IDS) Scheme" to detect the abnormal behavior (intrusion) of the subordinate (member) nodes and "Upwards-IDS Scheme" to detect the abnormal behavior of the cluster heads.

In order to detect intrusions towards WSNs, detecting the abnormal behaviors of the member nodes (in a cluster) is not sufficient. As mentioned in Chapter 4[1], after the clusters are formed and the CHs are elected, CHs may constitute a single point of failure. Therefore, in order to have a complete IDS for hierarchical WSNs, intrusions through CHs need to be detected as well.

It is important to emphasize that our focus in this research is on the hierarchical WSNs, meaning that sensor nodes are gathered into groups called "Clusters"[2]. Here, we would like to mention the references that are directly related to our proposal. In the IDS approaches proposed by [9], [10] and [11], the direction of the alert propagation is from sub-ordinates through CHs, leaving following question unanswered for the detection part: "What happens if a malicious CH drops the packet that is coming from a subordinate node and about to alert an upper level CH?". In the IDS approaches proposed by Agah *et al.* [12, 13], only one of the clusters of the network is monitored at a time. This leaves the rest of the network un-protected. In the IDS approach of Su *et al.* [14], both downwards and upwards protection are provided, meaning that CH's monitor subordinate nodes and vice versa,

[1]Readers that are interested to read more on IDSs and related work, may refer to Chapter 4.
[2]Readers that are interested to read more on clustering algorithms and related work, may refer to Chapter 5.

respectively. However, the proposed scheme uses SKC and therefore new nodes cannot be added to the network after the deployment, which makes the proposed scheme impracticable to be used.

For our IDS framework, we adopt the idea of downwards and upwards protection proposed by Su *et al.* [14]. For our D-IDS scheme, we adopted the "Isolation Table" concept that was suggested by Chen *et al.* [10] and also "Watchdog" concept that was suggested by Krontiris *et al.* [129]. For our U-IDS scheme, we adopted the "Monitoring Group" concept that was presented by Su *et al.* [14]. Finally, for both D-IDS and U-IDS schemes, as a detection algorithm, we adopt the "Sequential Probability Ratio Test" algorithm that was proposed by Brown and Du [130].

The rest of the chapter is organized as follows: Section 6.2 present the system model of the proposed IDS framework. Sections 6.3 and 6.4 presents the details of the D-IDS and U-IDS schemes, respectively. Details of the "Sequential Probability Ratio Test" that is used in our proposed D-IDS and U-IDS schemes is presented in Section 6.5, whereas Section 6.6 summarized the decision making process for each scheme. In Section 6.7, the effect of cluster size (maximum hops between cluster head and cluster members) on the detection probability of our Intrusion Detection System is investigated, when the IDS is located on the CH (D-IDS). In the reverse manner, in Section 6.8, the effect of total number of monitoring nodes on the detection probability of a malicious cluster head is investigated, when the IDS is located on the member nodes of a cluster (U-IDS). Finally, Section 6.9 concludes the chapter.

6.2 System Model

As the name implies, the proposed IDS is based on multi-level clustering; meaning that level-1 cluster heads (CHs) are the CHs for sensor nodes and at the same time subordinates for level-2 CHs, in the same manner, level-2 CHs are the CHs for level-1 CHs and at the same time subordinates for level-3 CHs, and so on, as shown in Figure 6.1.

For the selection of the cluster heads, our proposed power and connectivity aware clustering algorithm (for details, refer to Section 5) can be used as follows:

- Level-1 CH's would be selected by selecting the maximum hop size as "1".
- Level-2 CH's would be selected by selecting the maximum hop size as "2".
- Level-n CH's would be selected by selecting the maximum hop size as "n".

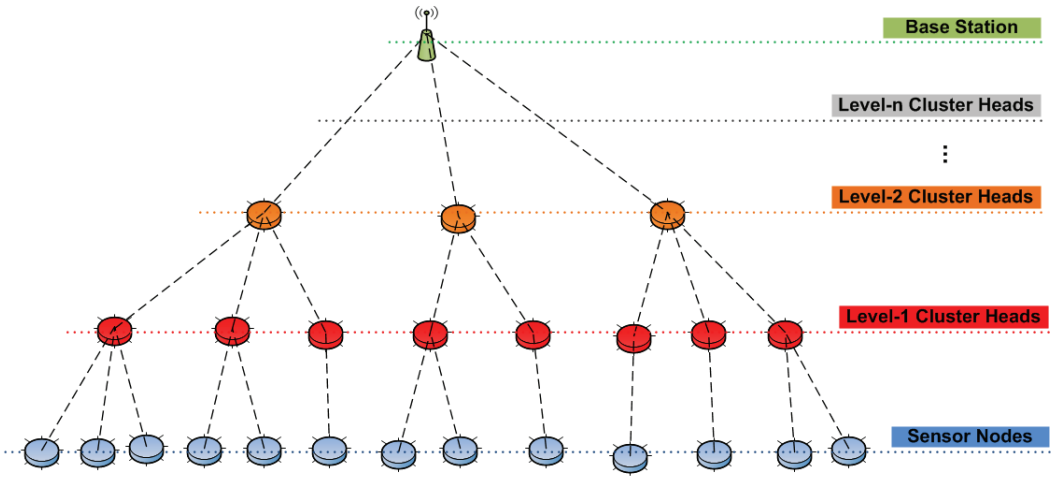

Figure 6.1 Multi-level clustering for our proposed IDS framework.

Our proposed IDS framework provides two types of intrusion detection approaches for the presented multi-level clustered WSN:

1. *Downwards Intrusion Detection System (D-IDS):* CH's are responsible for watching all the activities of their subordinates by using watchdogs and recording their activities in a table called "Isolation Table".

2. *Upwards Intrusion Detection System (U-IDS):* A certain number of (monitoring group size, m) subordinates coordinately monitor the activity of the CH and report any abnormal activity to an upper level CH.

Intrusions through subordinates of the network are detected by the D-IDS and intrusions through CHs of the network are detected by the U-IDS. By this way, our overall proposed IDS framework (D-IDS and U-IDS) covers entire network in terms of detecting intrusions.

6.3 Downwards Intrusion Detection System (D-IDS)

CHs hold watchdog counters with abnormality counters for each subordinate. Since the intrusion detection direction is from CH's towards subordinates, we call this scheme as "Downwards Intrusion Detection System (D-IDS)" For example, consider the network show in Figure 6.2. Here, Node-A is the level-1 CH of the remaining nodes. Therefore it has a watchdog (abnormality) counter for each subordinate node, namely, Node-1, Node-2, ..., Node-6.

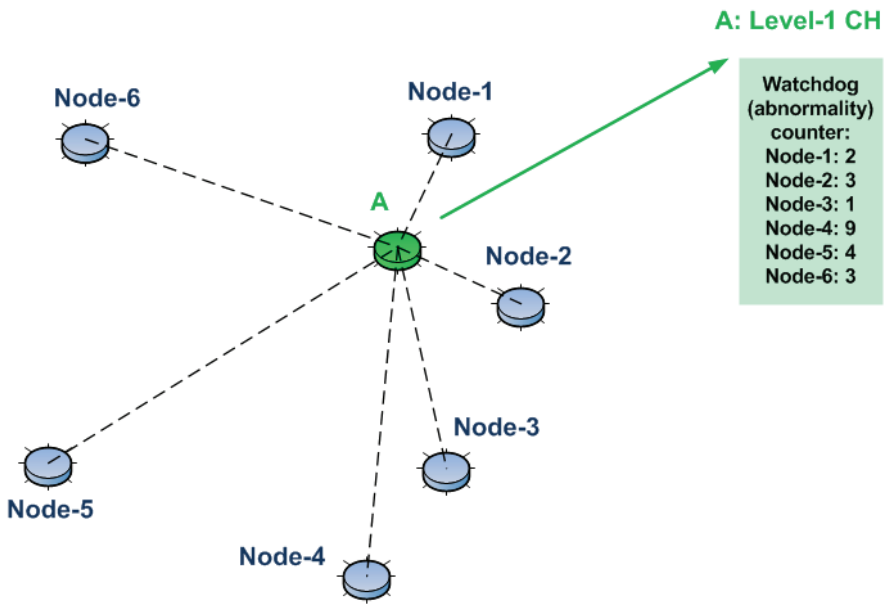

Figure 6.2 Usage of watchdog counters for our D-IDS.

Whenever any watchdog counter reaches a certain threshold, the associated node is flagged and included in the isolation table. Then as a mitigation step, any communication with this node is blocked (packets to be forwarded to this node as well as the packets coming from this node are dropped). For example, consider again the same network shown in Figure 6.2. But this time, assume that the watchdog counter of Node-4 is "10" and also assume that the threshold level for the watchdog counters is "10" as well. Since the watchdog counter of Node-4 has reached the threshold, Node-4 is marked as an abnormal node in the isolation table as shown in Figure 6.3. Then finally, all communications with Node-4 is blocked by the level-1 CH (Node-A).

The mentioned D-IDS is applicable to all levels. For example, consider the network shown in Figure 6.4. This time, Node-A is a subordinate of Node-X (an upper level node, level-2), just like Node-B and Node-C. Node-X holds watchdog counters for the abnormal behaviors of Node-A, Node-B and Node-C. As mentioned above, if any watchdog counter reaches a certain threshold, associated entry in the isolation table will be marked and a mitigation technique is issued (revocation of the node from the network).

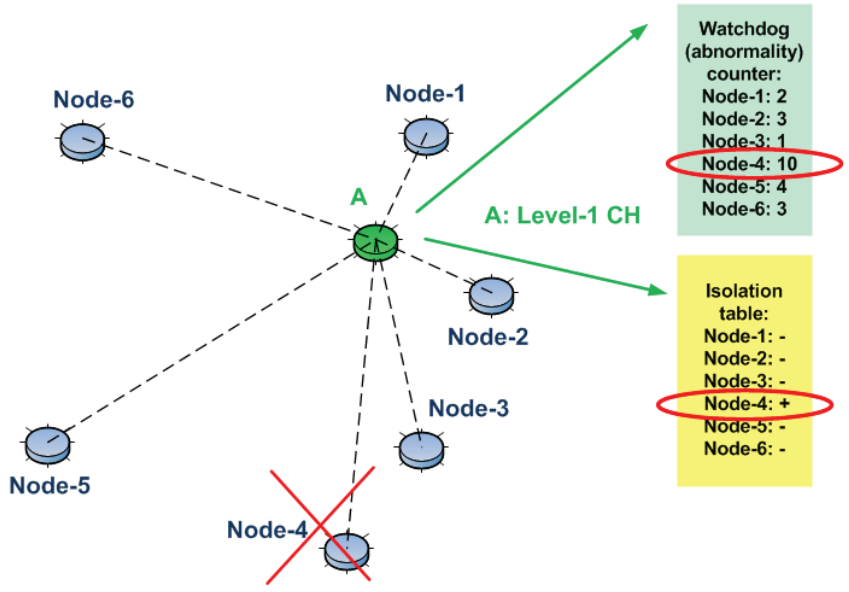

Figure 6.3 Usage of isolation table for our D-IDS.

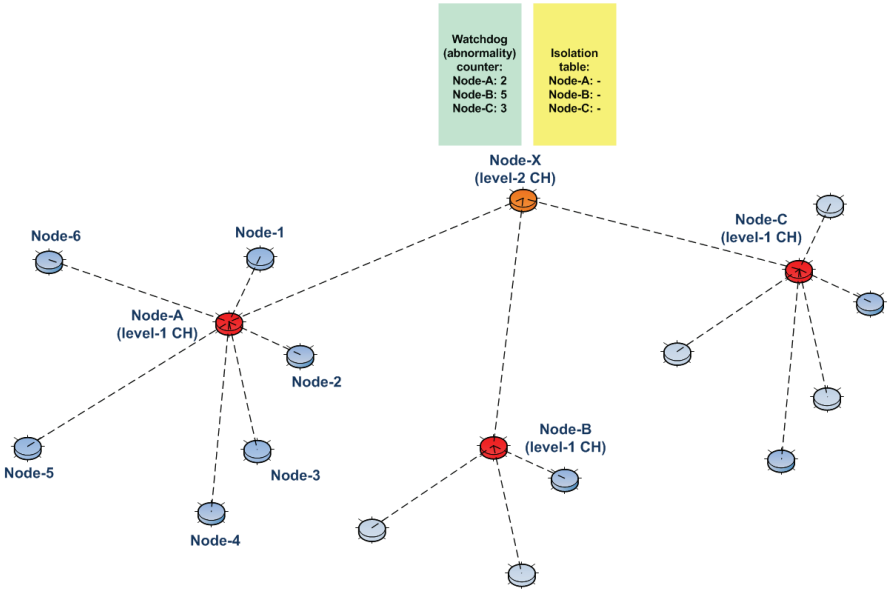

Figure 6.4 Implementation of D-IDS for upper levels of the network.

6.4 Upwards Intrusion Detection System (U-IDS)

A certain number of (monitoring group size, m) subordinates coordinately monitor the activity of the CH and report any abnormal activity to an upper level CH. Abnormal activity is determined when the watchdog counter reaches or exceeds a certain threshold. In accordance with the coordination concept, the total value of the abnormal cases is calculated by the logical "OR (+)" operation, individual watchdog results of each monitoring node is OR'ed with the rest of the watchdog results. As a result, the final decision associated with the abnormal behavior of the CH is concluded by the coordinated effort of all monitoring nodes.

The rationale for using OR operation is as follows: In some specific time interval, some of the monitoring nodes might be in "sleep" mode and therefore might miss the abnormal behavior of the CH. But in that specific time frame, the other monitoring nodes possibly would be in "awake" mode and catch the incidence. So, after a certain period of time (update interval), each monitoring node sends its' individual result to the rest of the monitoring nodes and final decision is made.

In order to catch most of the incidences (high probability of detection), the sleep/awake cycles of the monitoring nodes should be assigned accordingly. For instance, if there are 3 monitoring nodes in a cluster and if one of them is in sleep mode at a specific time frame, then in order to catch the incidences, the rest of the monitoring nodes should be in awake mode.

Consider the cluster of a network shown in Figure 6.5. Here, Node-A is again the level-1 CH of the remaining nodes. Subordinate nodes are, namely, Node-1, Node-2, ... Node-5 and Node-6. Among those, Node-1, Node-3 and Node-5 constitute the monitoring group. Their responsibility is to monitor the abnormal activity of Node-A (consider that Node-A is showing abnormal behavior for all time frames t_1, t_2, t_3, t_4). At a specific time frame, if the monitoring nodes are in awake mode and detect any abnormality, they update their watchdog counters, accordingly. For example, at the specific time frame of t_1, Node-1 was in sleep mode and therefore it was not able to detect any abnormality. But, Node-3 and Node-5 were in awake mode and detected an abnormality of Node-A and updated their own watchdog counters, accordingly (associated to the time frame of t_1). At the end of the time frame t_4, it is observed that out of 4 instances, Node-1 detected 2 instances, Node-3 detected 3 instances and Node-5 detected 2 instances.

The monitoring nodes share the entries of their watchdog counters among each other in certain time intervals. In the specific case shown in Figure 6.5, assume that the monitoring nodes share the

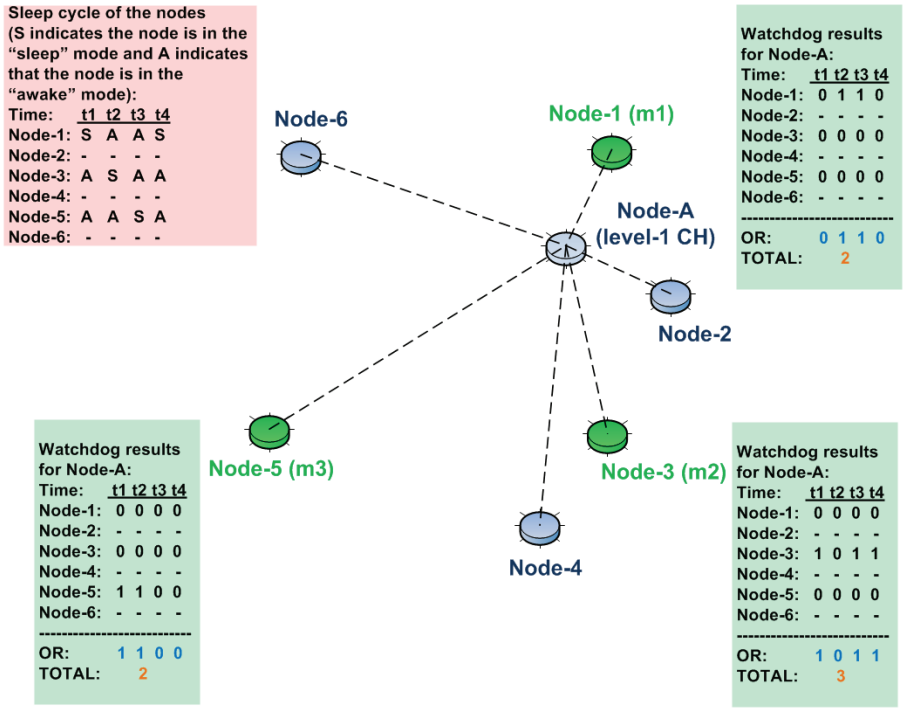

Figure 6.5 Usage of monitoring group for our U-IDS.

entries of their watchdog counters after the time frame t_4. This is shown in Figure 6.6. Here, as a result of the arrival of the watchdog counter entries from Node-3 and Node-5, Node-1 ($m1$) updates the watchdog counter entries associated with them and finally the total number of encounters is calculated by the OR operation as mentioned earlier. In accordance with the updates from Node-3 and Node-5, Node-1 has refreshed its watchdog counter and calculated the total number of incidences as "4". In the same manner, Node-3 and Node-5 will update their watchdog counter entries and calculate the total number of incidences as "4".

After each update interval, monitoring nodes check their updated counter values. Whenever their watchdog counter reach a certain threshold ("15" in our example), they send an encrypted alert message to an upper level CH. The reason of encryption is to hide the alert message from the CH that is under investigation (Node-A in our example).

These alert messages are sent directly to the upper level cluster head. In order to do so, we assume that the radios of the nodes have two different operation modes: "Normal" and "Alert". In the normal mode, since CH's are generally in one hop away, the radios operate to transmit in short range. This energy saving mode helps nodes to increase their life-time. In the alert mode, the

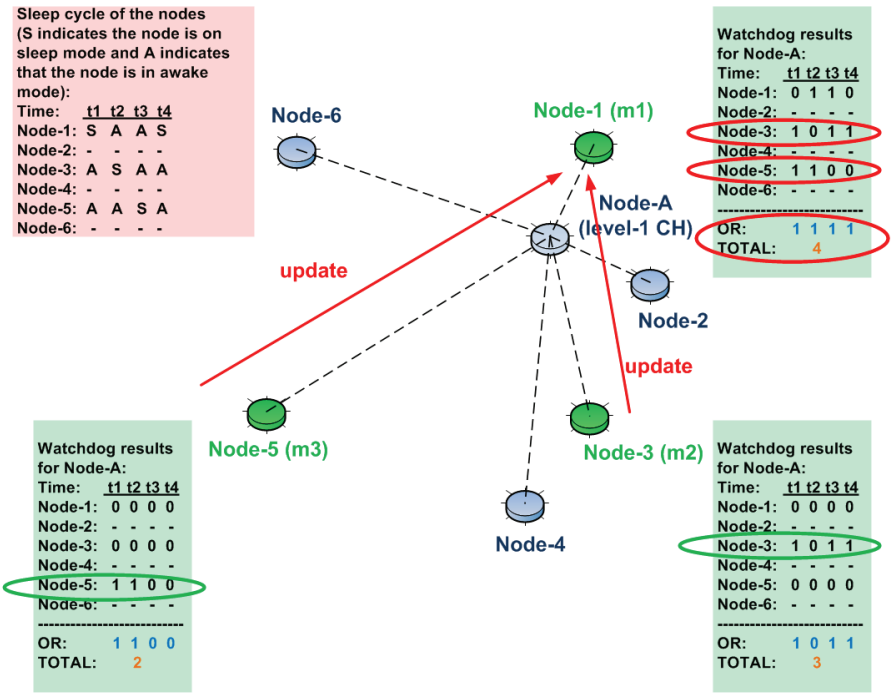

Figure 6.6 Watchdog update propagation in our U-IDS.

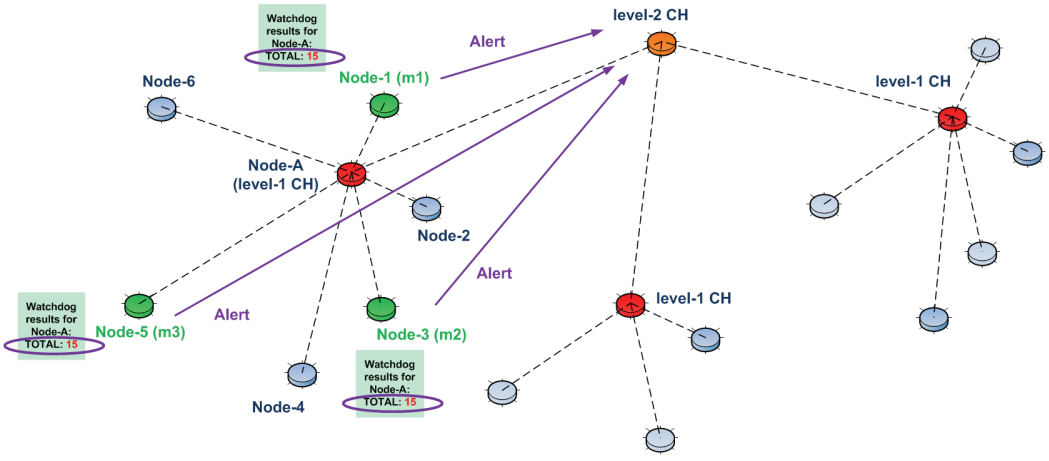

Figure 6.7 Alert propagation towards upper levels in our U-IDS.

radios operate to transmit in long range. By this way, without the help of intermediate nodes, a monitoring node can directly send the alert messages to an upper level CH.

Consider the case shown in Figure 6.7. Here, as mentioned above, in each update interval, each monitoring node updates the other monitoring nodes and also check its' watchdog counter. After a certain period of time, watchdog counters of all the monitoring nodes reached the threshold value of "15". Therefore, they changed their radio's mode of operation to "Alert" mode and sent an encrypted alert message directly to an upper level CH, namely a level-2 CH. In this specific example, Node-1 has a higher probability to alert level-2 CH, since its location is closer than Node-3 and Node-5.

6.5 Detection of DoS Attacks in WSNs by using Sequential Probability Ratio Test

Wald's Sequential Probability Ratio Test (SPRT) for detection of Selective Forwarding Attacks (a DoS attack in network layer of WSNs) was used in [130]. This method detects intentional packet drops with a high probability of detection rate. Therefore, it can be applied to any packet drop attack towards the security of WSNs, for example DoS attacks in network layer (blackhole attacks, sinkhole attacks, etc.).

According to SPRT, a random variable p is used to define the status of packet forwarding, where 0 denotes successful transmission of the packet (Good), and 1 denotes a packet drop (Bad). p is calculated as the percentage of dropped packets over all packets to be transmitted. p' is defined as the acceptable probability of dropped packets. A node is considered as legitimate if $p \leq p'$ holds, and it is considered as compromised if $p > p'$ holds. $p_0 < p' < p_1$ defines the "gray region" for the decision making, where the decision is inconclusive regarding the legitimacy of the node. Figure 6.8 pictures the decision boundaries for the SPRT, namely; white, gray and black regions.

Note that $1, 2, \ldots, m$ represents the sample number. The ultimate goal is to minimize the miss detection rate, $\alpha = P_1(|D_m = 0)$ and the false alarm rate, $\beta = P_0(|D_m = 1)$, where D_m stands for the decision at step m. At the same time, we need to achieve this in minimum number of samples (m_{min}). SPRT calculates this number as shown in Equation 6.1:

$$m_{min} = \frac{L(p)\log(\beta) + (1 - L(P))\log(\alpha)}{p\log(\frac{p_1}{p_0}) + (1-p)\log(\frac{1-p_1}{1-p_0})} \tag{6.1}$$

Where $L(p)$ is calculated as shown in Equation 6.2,

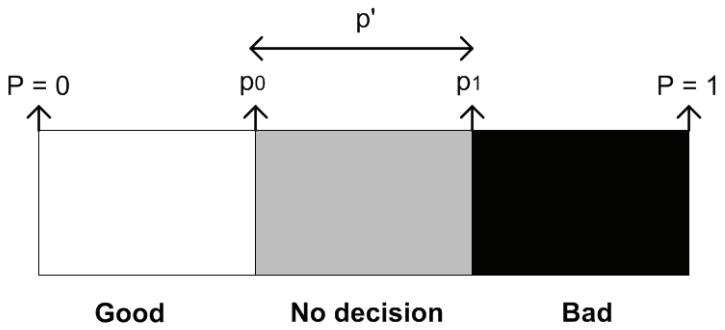

Figure 6.8 Thresholds for the decision making.

$$L(p) = \frac{(\frac{1-\beta}{\alpha})^h - 1}{(\frac{1-\beta}{\alpha})^h - (\frac{\beta}{1-\alpha})^{-h}} \tag{6.2}$$

and h can be determined by solving the Equation 6.3:

$$p = \frac{1 - (\frac{1-p_1}{1-p_0})^h}{(\frac{p_1}{p_0})^h - (\frac{1-p_1}{1-p_0})^h} \tag{6.3}$$

After setting all the parameters $(p_0, p_1, \alpha, \beta)$ and collecting m samples, the acceptance threshold (a_m) and the rejection threshold (r_m) can be found by using Equations 6.4 and 6.5, respectively:

$$a_m = \frac{\log(\frac{\beta}{1-\alpha})}{\log(\frac{p_1}{p_0}) - \log(\frac{1-p_1}{1-p_0})} + m\frac{\log(\frac{1-p_0}{1-p_1})}{\log(\frac{p_1}{p_0}) - \log(\frac{1-p_1}{1-p_0})} \tag{6.4}$$

$$r_m = \frac{\log(\frac{1-\beta}{\alpha})}{\log(\frac{p_1}{p_0}) - \log(\frac{1-p_1}{1-p_0})} + m\frac{\log(\frac{1-p_0}{1-p_1})}{\log(\frac{p_1}{p_0}) - \log(\frac{1-p_1}{1-p_0})} \tag{6.5}$$

Let $\lceil a_m \rceil$ denotes upper bound for a_m and $\lfloor r_m \rfloor$ denotes lower bound for r_m. For each sample, $\lceil a_m \rceil$ and $\lfloor r_m \rfloor$ should be revised according to new values of a_m and r_m respectively.

Let d_m denotes the number of packets dropped in the m number of samples. Then, SPRT test needs to be continuously performed as long as the Equation 6.6 holds:

$$\lceil a_m \rceil < d_m < \lfloor r_m \rfloor \tag{6.6}$$

At each round of the SPRT test, there are three possible outcomes based on the result of the comparison shown in Equation 6.6:

1. If $d_m \leq \lceil a_m \rceil$, then the conclusion is that the node is legitimate.
2. If $\lceil a_m \rceil < d_m < \lfloor r_m \rfloor$, then the SPRT test needs to be continued.
3. $\lfloor r_m \rfloor \leq d_m$, then the conclusion is that the node is compromised.

6.6 Decision Making in IDSs

According to Patcha *et al.* [74], decision engine (i.e., decision making algorithm) of an IDS concludes either one of four decisions (with non-zero probabilities) as a result of the decision making process over a triggered alarm (event):

- Intrusive but not anomalous (false-negative): There is an intrusion to the system, but IDS fails to detect it and concludes the event as non-anomalous one.
- Not intrusive but anomalous (false-positive): There is no intrusion to the system, but IDS mistakenly concludes a normal event as an anomalous one.
- Not intrusive and not anomalous (true-negative): There is no intrusion to the system, and IDS concludes the event as non-anomalous one.
- Intrusive and anomalous (true-positive): There is an intrusion to the system, and IDS concludes the event as an anomalous one.

Figure 6.9 summarized the mentioned possibilities regarding the legitimacy assessment of a node. Eventually we expect the selected decision making algorithm to generate more percentage of true-positives and true-negatives, and less percentage of false-positives and false-negatives.

6.7 Effect of Cluster Size on the Detection Probability of the D-IDS

In this section, we followed Shin *et al.*'s approach [9] to evaluate the effect of clustering on the detection probability of D-IDS. Our main focus is to calculate the effect of the maximum distance (hops) between cluster head (CH) and cluster members on the intrusion detection probability.

In Figure 6.10, a WSN is divided into clusters. Maximum distance of each cluster is 2 hops, meaning that in a cluster, a CH (denoted as red nodes in the figure) is maximum of 2 hops away

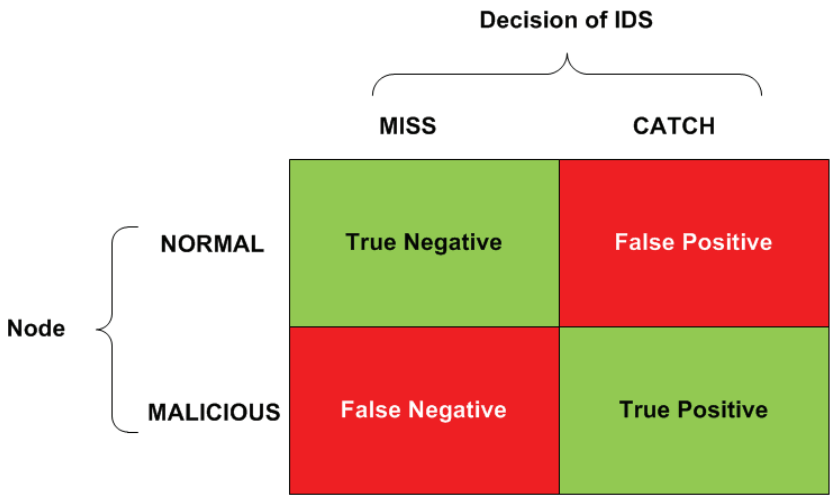

Figure 6.9 All possible detection results of an IDS.

from its member nodes. As an example; node A is a CH and nodes B and C are its member nodes. 2-hop cluster does not mean that node C is in direct communication range of A (as we can see in the figure, C is outside of the communication range of A), but it means that A can connect and monitor behavior of C through node B (since B is in communication range of both A and C, B performs as a relaying node between nodes A and C).

CHs (such as node A) can use indirect monitoring (A monitors node C through node B) through intermediate nodes (such as node B) to detect intrusions that would happen at the end nodes (such as node C). However, this kind of monitoring definitely will increase the network overhead. Besides, reliability of the intermediate nodes, such as the sleep rate and error rate, would certainly change the overall performance of the intrusion detection system.

Let us define the average *Sleep Rate* and the average *Error Rate*[3] of all the nodes (except CH and the malicious node) to be s and e, respectively. Here, we assume that the probability distributions of these random variables (s and e) to be Gaussian with means $0 \leq E(s) \leq 1$ and $0 \leq E(e) \leq 1$; variances σ_s^2 and σ_e^2.

If a CH detects a malicious node located at j-hop away from the CH in an i-hop cluster, where i is the cluster size in terms of number of hops, and if $0 \leq j \leq i$ holds; then the *Detection Probability* ($p_{i,j}$) of each node is given by:

[3]Here, *Error Rate* is due to packet losses caused by the transmission problems, such as packet collisions.

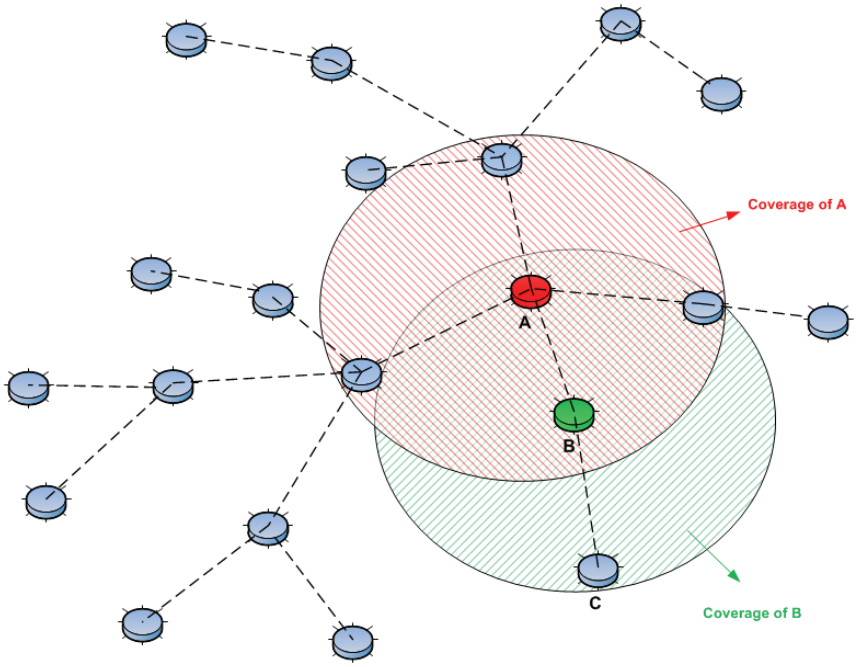

Figure 6.10 An example of clustered network with a maximum hop distance of 2.

$$p_{i,j} = \{(1 - E(s))(1 - E(e))\}^{j-1} \tag{6.7}$$

In Table 6.1, different values of $p_{i,j}$ are shown as i and j are varied[4].

Table 6.1 Detection probability ($p_{i,j}$) for different values of i and j.

	$j = 1$	$j = 2$	$j = 3$	$j = 4$
$i = 1$	1	N/A	N/A	N/A
$i = 2$	1	$(1 - E(s))(1 - E(e))$	N/A	N/A
$i = 3$	1	$(1 - E(s))(1 - E(e))$	$\{(1 - E(s))(1 - E(e))\}^2$	N/A
$i = 4$	1	$(1 - E(s))(1 - E(e))$	$\{(1 - E(s))(1 - E(e))\}^2$	$\{(1 - E(s))(1 - E(e))\}^3$

Finally, the *Average Detection Probability* (P_i) of i-hop cluster is given by:

$$P_i = \frac{1}{i}\sum_{j=1}^{i} p_{i,j} = \frac{1}{i}\sum_{j=1}^{i} \{(1 - E(s))(1 - E(e))\}^{j-1} \tag{6.8}$$

[4]Here, note that the value of j cannot exceed the value of i.

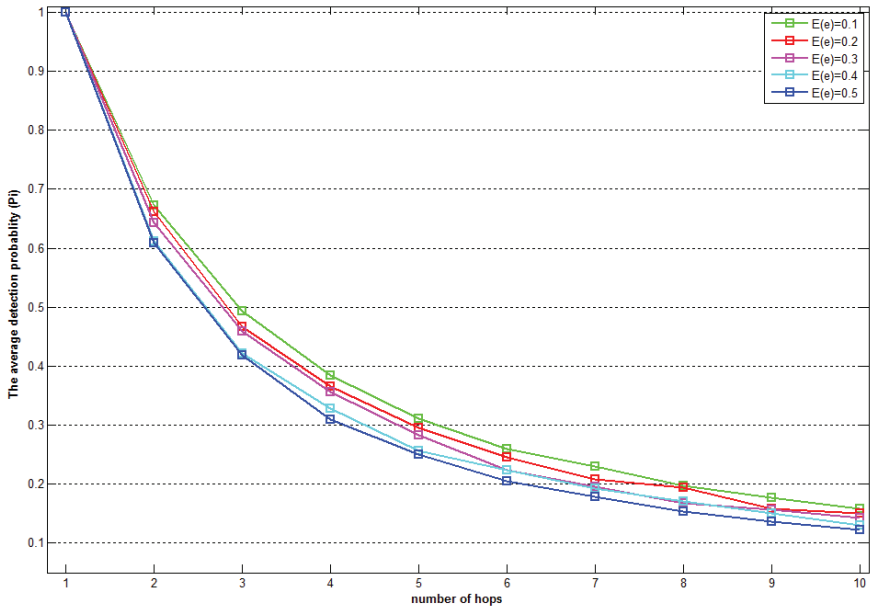

Figure 6.11 Effect of cluster size on the detection probability of the D-IDS for various packet loss rates while the sleep rate is 60%.

Figures 6.11 and 6.12 show different values of P_i as the maximum number of hops (i) changes according to Equation 6.8. There are 5 plots in each figure, for various values of $E(e)$ in Figure 6.11 and $E(s)$ in Figure 6.12, respectively.

In Figure 6.11, $E(s) = 0.6$ means that the average sleep rate of the nodes is 60%. Each plot represents a different value of $E(e)$ with a variance of 10%. From the plots, it can be observed that as the packet loss rate of each node increases, the average intrusion detection probability of the D-IDS decreases, which is expected. As the error rate increases, it becomes difficult for the D-IDS to determine if the loss of a packet was caused by a channel error (natural causes) or an outside (intruder) effect.

In Figure 6.12, $E(e) = 0.3$ means that the average packet drop rate of the nodes is 30%. Each plot represents a different value of $E(s)$ with a variance of 10%. From the plots, it can be observed that as the sleep rate of each node increases, the average intrusion detection probability of the D-IDS decreases, which is also expected. As the sleep rate increases, the chance of the D-IDS for catching an intrusion decreases. Hence, the fewer number of nodes are awake in the network, the lesser intrusions will be caught by the D-IDS. In other words, an intrusion that would be caught by

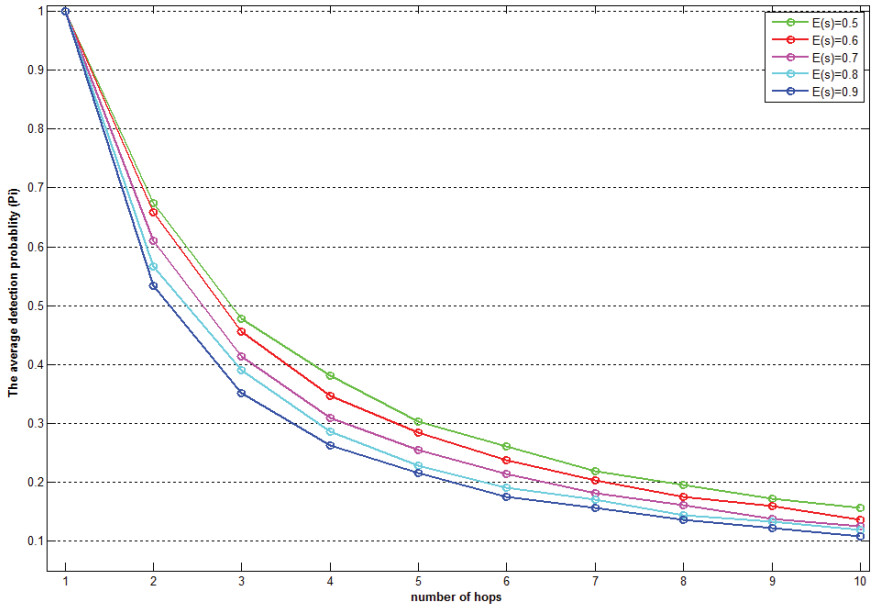

Figure 6.12 Effect of cluster size on the detection probability of the D-IDS for various sleep rates while the packet loss rate is 30%.

an intermediate node would be simply missed, because the node was sleeping at the exact time that the intrusion happened.

From the plots of Figures 6.11 and 6.12, we can conclude that as the number of maximum range of a cluster (maximum number of hops from a CH and a member node in a cluster) increases, the average intrusion detection probability of the D-IDS decreases. In other words, indirect monitoring of the intrusions has a bad effect on the overall intrusion detection probability of the IDS.

6.8 Effect of Monitoring Group Size on the Detection Probability of the U-IDS

In this section, we follow Yang *et al.*'s work [131], to investigate the effect monitoring group size (m) on the detection probability of malicious cluster heads in our U-IDS. Here, we assume that m of the member nodes of a cluster has an intrusion detection scheme that are running in a collaborative manner. Thus, cluster members periodically give a decision regarding the trustworthiness of a cluster head.

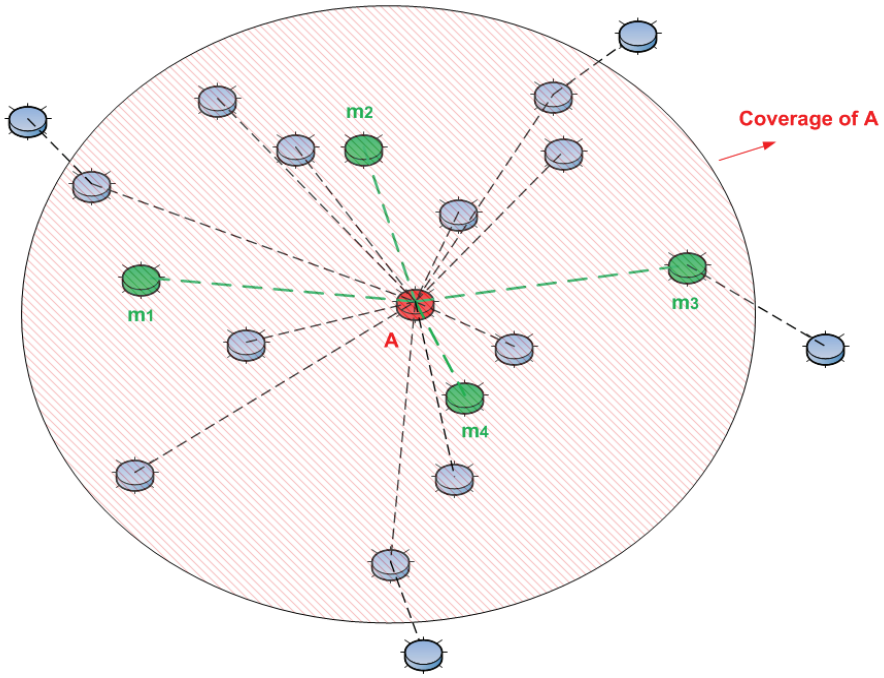

Figure 6.13 A typical 15-node clustered WSN (1-hop distance).

Consider the clustered WSN (1-hop distance) shown in Figure 6.13. Node A (marked with red color) is the cluster head and it has 15 member nodes, which are in the radio coverage of A. Among these member nodes, 4 of them (m=4; marked with green color and denoted as m_1,m_2,m_3 and m_4) are collaborating to monitor the activity of A.

In order to calculate the effect monitoring group size (m) on the detection probability of malicious cluster head, assume that the size of a cluster is N (15 in our case) and the probability of each member node to detect the malicious cluster head is P_d. Then, the total probability of the malicious cluster head to be detected P_D, as a result of the collaboration size (m) (m number of the nodes) is calculated as shown in 6.9:

$$P_D = \sum_{k=m}^{N} \binom{N}{k} P_d^k (1 - P_d)^{N-k} \tag{6.9}$$

Figure 6.14 shows the change of overall detection probability (P_D) with respect to the collaboration size (m number of the nodes) for various values of P_d, for a cluster size of 15 ($N = 15$). Accordingly, it can be observed that collaboration is useful in increasing the overall detection rate

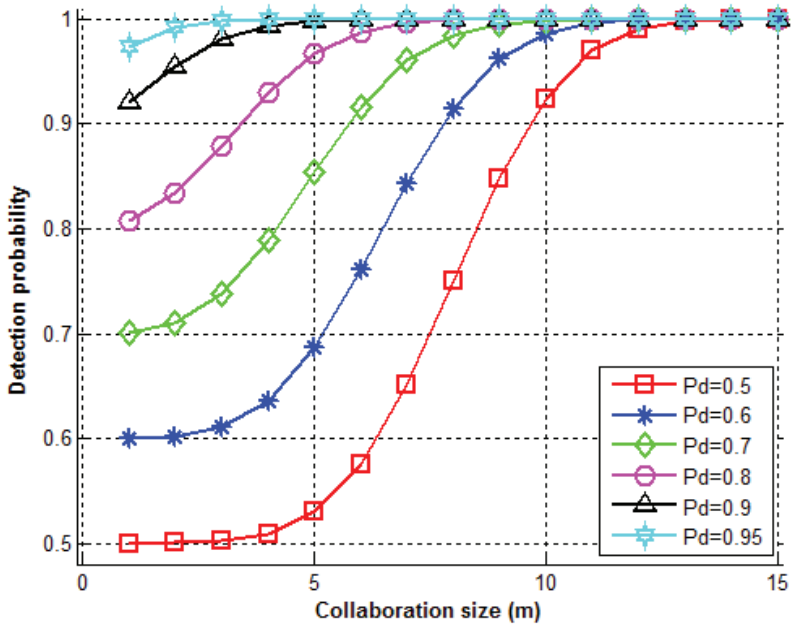

Figure 6.14 Detection probability (P_D) vs. collaboration size(m) for various values of P_d.

(P_D). As the collaboration size (monitoring group size, m) increases, overall detection probability increases and approaches to 1 (100%).

Again, assume that the size of a cluster is N and the probability of each member node to fail (false-alarm) in detecting the malicious cluster head is P_f. Then, the total probability of the malicious cluster head goes un-detected (false-alarm) P_F, as a result of the collaboration size (m number of the nodes) is calculated as shown in 6.10:

$$P_F = \sum_{k=m}^{N} \binom{N}{k} (1 - P_f)^k (P_f)^{N-k} \tag{6.10}$$

Figure 6.15 shows the change of overall false-alarm probability (P_F) with respect to the collaboration size (m number of the nodes) for various values of P_f, for a cluster size of 15 ($N = 15$). Accordingly, it can be observed that collaboration is useful in decreasing the overall false-alarm rate (P_F). As the collaboration size (monitoring group size, m) increases, overall false-alarm probability decreases and approaches to 0 (0%).

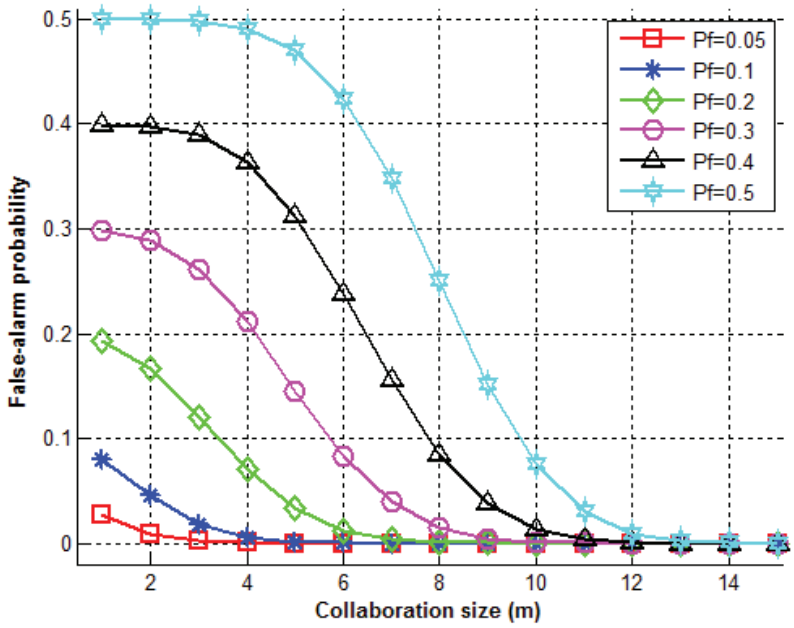

Figure 6.15 Detection probability (P_F) vs. collaboration size(m) for various values of P_f.

6.9 Conclusions and Suggestions for Future Research

This chapter presented our proposed Intrusion Detection System (IDS) framework based on multi-level clustering for hierarchical wireless sensor networks. It is based upon the proposed clustering algorithm (the nodes use our proposed clustering algorithm while forming their clusters, see Chapter 5). Our proposed IDS framework provides two types of intrusion detection approaches, namely "Downwards-IDS" to detect the abnormal behavior (intrusion) of the subordinate (member) nodes and "Upwards-IDS" to detect the the abnormal behavior of the cluster heads.

The effect of cluster size (maximum hops between cluster head and cluster members) on the detection probability of a malicious node was evaluated, when the IDS is located on the CH (Downwards-IDS). In the same manner, the effect of total number of monitoring nodes on the detection probability of a malicious cluster head was evaluated, when the IDS is located on the member nodes of a cluster (Upwards-IDS).

There is a trade-off between "maximum hop count" and "intrusion detection probability". As the maximum hop count increases, intrusion detection probability (of an IDS) decreases and vice versa. According to the results of the analytical calculations presented in Section 6.7, following

recommendations are provided for the maximum hop count: Figures 6.11 and 6.12 suggest to keep the maximum hop distance lower than "4". Maximum hop distance should be selected as "2" or "3", depending on the "sleep rate" of the nodes and "average packet loss rate" of the network (1.0 represents 100% probability of a node to be sleeping or a packet to be lost): If the "sleep rate" and/or "average packet loss rate" are higher than 0.7; then the maximum hop distance should be selected as "2", otherwise it should be selected as "3".

As in most technologies, nothing comes for free. By using more number of monitoring members, higher detection rates and lower false alarm rates can be achieved. The cost for this achievement is the loss of scarce resources (e.g., energy). Therefore, a proper trade-off point need to be determined in finding the right number for the monitoring group size (m).

As Figure 6.14 in Section 6.8 suggests, out of 15 nodes in each cluster, by selecting m=7; very satisfactory detection probability ($> 95\%$) can be achieved if the individual detection probabilities are higher than 70%. Again, out of 15 nodes in each cluster and for the same group size (m=7), Figure 6.15 suggests that the false-alarm probability will be lower than 5% if the individual false-alarm rates are lower than 30%.

CHAPTER 7 :

CONCLUSION AND FUTURE WORK

7.1 Conclusions

In order to protect Wireless Sensor Networks (WSNs) from intrusions (attacks), this dissertation presents a security provisioning plan (please refer to Chapter 2) that consists of three main components: 1)Prevention, 2)Detection, and 3)Mitigation, of intrusions. Solutions to the first two components of the security provisioning plan are proposed in this dissertation: an Intrusion Prevention System (IPS) and an Intrusion Detection System (IDS).

The proposed IPS scheme (please refer to Chapter 3) targets intrusion prevention in user level; whereas the proposed IDS framework (please refer to Chapter 6) targets intrusion detection in both sensor level and CH level.

The proposed IPS scheme employs both PKC and SKC approaches, so that it takes advantage of both schemes. Analysis and simulation results have shown that, the proposed IPS scheme is not only more secure and yet scalable than existing SKC based schemes, but also requires lesser processing power and provides higher energy efficiency than existing PKC based schemes. Proposed IPS scheme brings advantages (scalability, flexibility) of PKC, without requirement of extra cost (in terms of energy) on the sensor nodes. Besides, time cost of the proposed IPS scheme is very negligible compared to the existing PKC based schemes.

The proposed IDS framework provides two types of intrusion detection approaches, namely "Downwards-IDS" to detect the abnormal behavior (intrusion) of the subordinate (member) nodes and "Upwards-IDS" to detect the the abnormal behavior of the cluster heads. The effect of cluster size (maximum hops between cluster head and cluster members) on the detection probability of a malicious node was evaluated, when the IDS is located on the CH (Downwards-IDS). Similarly, the effect of total number of monitoring nodes on the detection probability of a malicious cluster head was evaluated, when the IDS is located on the member nodes of a cluster (Upwards-IDS). Following the evaluations, optimum numbers for the mentioned parameters are suggested.

For both components of the security plan (IPS and IDS), clustering is a requirement; meaning that after deployment, sensor nodes form clusters and elect cluster heads. The proposed power and connectivity aware clustering algorithm (please refer to Chapter 5) is the main workhorse in achieving this.

According to the energy consumption simulation results, our proposed power and connectivity aware clustering algorithm out performed existing clustering algorithm in the literature, in terms of energy efficiency and also total life-time of the network.

7.2 Future Work

The testing and performance evaluation of our proposed IDS framework for a specific attack, such as blackhole attack, is left as a future work. Besides, in order to inspect the efficiency of the security provisioning plan for WSNs, the interaction and behavior of the proposed IPS scheme and IDS framework have to be investigated while they are operating together.

In order to investigate real time performances of the proposed algorithms, it is worth considering the hardware implementation with real sensor devices. Although we have done some simple implementation tests on 2-3 sensor nodes, it would be better to observe the behavior of our algorithms while they are operating on a larger scale network consisting of 30-50 sensor nodes.

In general, nodes in the WSNs are considered to be stationary. So, throughout this dissertation, our proposed algorithms are evaluated accordingly. But for some special applications of mobile WSNs, the effects of mobility on the proposed algorithms have to be studied.

We anticipate that in providing energy efficient clustering and prevention/detection of intrusions, the proposed algorithms and schemes presented in this dissertation can be applied (with some modifications) to the new emerging technologies such as *pervasive computing*, *cloud computing*, *ubiquitous computing* and *internet of things*.

REFERENCES

[1] D. Kushner, "The real story of stuxnet," *IEEE Spectrum*, vol. 50, no. 3, pp. 49–53, 2013.

[2] A. Perrig, R. Szewczyk, V. Wen, D. Culler, and J. Tygar, "Spins: security protocols for sensor networks." Rome, Italy,: 7th annu. Int. conf. on mobile computing and networking (MOBICOM01), August 2001, pp. 189–199.

[3] C. Karlof, N. Sastry, and D. Wagner, "Tinysec: a link layer security architecture for wireless sensor networks," in *Proceedings of the 2nd international conference on Embedded networked sensor systems.* ACM, 2004, pp. 162–175.

[4] S. Zhu, S. Setia, and S. Jajodia, "Leap: Efficient security mechanisms for large-scale distributed sensor networks," *Computer and communications security*, 2003.

[5] K. Wong, Y. Zheng, J. Cao, and S. Wang, "A dynamic user authentication scheme for wireless sensor networks," in *Sensor Networks, Ubiquitous, and Trustworthy Computing, 2006. IEEE International Conference on*, vol. 1. IEEE, 2006, pp. 8–pp.

[6] H. Tseng, R. Jan, and W. Yang, "An improved dynamic user authentication scheme for wireless sensor networks," in *Global Telecommunications Conference, 2007. GLOBECOM'07. IEEE.* IEEE, 2007, pp. 986–990.

[7] X. Le, S. Lee, and Y. Lee, "Two-tier user authentication scheme for heterogeneous sensor networks," in *the 5th IEEE International Conference on Distributed Computing in Sensor Systems,(DCOSS09), Marina Del Rey, California, USA*, 2009.

[8] Z. Benenson, N. Gedicke, and O. Raivio, "Realizing robust user authentication in sensor networks," in *Workshop on Real-World Wireless Sensor Networks (REALWSN)*, 2005.

[9] S. Shin, T. Kwon, G. Jo, Y. Park, and H. Rhy, "An experimental study of hierarchical intrusion detection for wireless industrial sensor networks," *Industrial Informatics, IEEE Transactions on*, vol. 6, no. 4, pp. 744–757, 2010.

[10] R. Chen, C. Hsieh, and Y. Huang, "A new method for intrusion detection on hierarchical wireless sensor networks," in *Proceedings of the 3rd International Conference on Ubiquitous Information Management and Communication.* ACM, 2009, pp. 238–245.

[11] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. Bezdek, "Distributed anomaly detection in wireless sensor networks," in *Communication systems, 2006. ICCS 2006. 10th IEEE Singapore International Conference on.* IEEE, 2006, pp. 1–5.

[12] A. Agah, S. Das, K. Basu, and M. Asadi, "Intrusion detection in sensor networks: A non-cooperative game approach," in *Network Computing and Applications, 2004.(NCA 2004). Proceedings. Third IEEE International Symposium on.* IEEE, 2004, pp. 343–346.

[13] A. Agah and S. Das, "Preventing dos attacks in wireless sensor networks: A repeated game theory approach," *International Journal of Network Security*, vol. 5, no. 2, pp. 145–153, 2007.

[14] C. Su, K. Chang, Y. Kuo, and M. Horng, "The new intrusion prevention and detection approaches for clustering-based sensor networks [wireless sensor networks]," in *Wireless Communications and Networking Conference, 2005 IEEE*, vol. 4. IEEE, 2005, pp. 1927–1932.

[15] O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on.* IEEE, 2003, pp. 8–pp.

[16] Q. Chen, J. Ma, Y. Zhu, D. Zhang, and L. Ni, "An energy-efficient k-hop clustering framework for wireless sensor networks," *Wireless Sensor Networks*, pp. 17–33, 2007.

[17] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-efficient communication protocol for wireless microsensor networks," in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on.* IEEE, 2000, pp. 10–pp.

[18] S. Bandyopadhyay and E. Coyle, "An energy efficient hierarchical clustering algorithm for wireless sensor networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3. IEEE, 2003, pp. 1713–1723.

[19] O. Younis and S. Fahmy, "Heed: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks," *Mobile Computing, IEEE Transactions on*, vol. 3, no. 4, pp. 366–379, 2004.

[20] M. Ye, C. Li, G. Chen, and J. Wu, "Eecs: an energy efficient clustering scheme in wireless sensor networks," in *Performance, Computing, and Communications Conference, 2005. IPCCC 2005. 24th IEEE International.* IEEE, 2005, pp. 535–540.

[21] J. Jia, Z. He, J. Kuang, and Y. Mu, "An energy consumption balanced clustering algorithm for wireless sensor network," in *Wireless Communications Networking and Mobile Computing (WiCOM), 2010 6th International Conference on.* IEEE, 2010, pp. 1–4.

[22] C. Li, M. Ye, G. Chen, and J. Wu, "An energy-efficient unequal clustering mechanism for wireless sensor networks," in *Mobile Adhoc and Sensor Systems Conference, 2005. IEEE International Conference on.* IEEE, 2005.

[23] M. Brust, A. Andronache, S. Rothkugel, and Z. Benenson, "Topology-based clusterhead candidate selection in wireless ad-hoc and sensor networks," in *Communication Systems Software and Middleware, 2007. COMSWARE 2007. 2nd International Conference on.* IEEE, 2007, pp. 1–8.

[24] I. Butun and R. Sankar, "A brief survey of access control in wireless sensor networks," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE.* IEEE, 2011, pp. 1118–1119.

[25] X. Le, S. Lee, I. Butun, M. Khalid, and R. Sankar, "An energy-efficient access control scheme for wireless sensor networks based on elliptic curve cryptography," *Journal of Communications and Networks*, vol. 11, no. 6, p. 599, 2009.

[26] I. Butun, Y. Wang, Y. Lee, and R. Sankar, "Intrusion prevention with two–level user authentication in heterogeneous wireless sensor networks," *International Journal of Security and Networks*, vol. 7, no. 2, pp. 107–121, 2012.

[27] I. Butun and R. Sankar, "Advanced two tier user authentication scheme for heterogeneous wireless sensor networks," in *Consumer Communications and Networking Conference (CCNC), 2011 IEEE.* IEEE, 2011, pp. 169–171.

[28] I. Butun, Y. Wang, and R. Sankar, "Evaluation of advanced two tier user authentication scheme," in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE*. IEEE, 2012, pp. 159–163.

[29] I. Butun, S. Morgera, and R. Sankar, "A survey of intrusion detection in wireless sensor networks," *IEEE Communications Surveys and Tutorials (revised version submitted)*, 2013.

[30] I. Butun and R. Sankar, "Power and connectivity aware clustering for wireless sensor networks," *Elsevier Journal of Computer Networks (submitted)*, 2013.

[31] ——, "An intrusion detection system based on multi-level clustering for hierarchical wireless sensor networks," *Elsevier Journal of Network Security (to be submitted)*, 2013.

[32] R. Rao and G. Kesidis, "Detecting malicious packet dropping using statistically regular traffic patterns in multihop wireless networks that are not bandwidth limited," in *Global Telecommunications Conference, 2003. GLOBECOM'03. IEEE*, vol. 5. IEEE, 2003, pp. 2957–2961.

[33] I. F. B. High Technology Business Development. (2012) Thinking big: $1 trillion mems market part 1.

[34] Y. Wang, G. Attebury, and B. Ramamurthy, "A survey of security issues in wireless sensor networks," *Communications Surveys and Tutorials, IEEE*, vol. 8, no. 2, pp. 2–23, 2006.

[35] Libelium. (2012) Wasp mote - wireless sensor networks 802.15.4 zigbee mote. [Online]. Available: http://www.libelium.com/products/waspmote

[36] Memsic. (2012) Micaz mote. [Online]. Available: http://www.memsic.com/products/wireless-sensor-networks/wireless-modules.html

[37] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *Communications Magazine, IEEE*, vol. 40, no. 8, pp. 102–114, 2002.

[38] Y. Zhang, H. Hu, and M. Fujise, *Resource, mobility, and security management in wireless networks and mobile communications*. Auerbach Publications, 2006.

[39] S. I. S. RLR UK. (2012) Pragmatic approach to security. [Online]. Available: http://blog.rlr-uk.com/2010/01/pragmatic-approach-to-security.html

[40] A. Al-Ayed, S. Furnell, D. Zhao, and P. Dowland, "An automated framework for managing security vulnerabilities," *Information management & computer security*, vol. 13, no. 2, pp. 156–166, 2005.

[41] C. Andrew, "The five ps of patch management is there a simple way for businesses to develop and deploy an advanced security patch management strategy?" *Computers & Security*, vol. 24, no. 5, p. 362, 2005.

[42] D. White, "Limiting vulnerability exposure through effective patch management: threat mitigation through vulnerability remediation," 2007.

[43] D. Zhao, S. Furnell, and A. Al-Ayed, "The research on a patch management system for enterprise vulnerability update," in *Information Engineering, 2009. ICIE'09. WASE International Conference on*, vol. 2. IEEE, 2009, pp. 250–253.

[44] I. Butun, M. Al-Faruque, and L. Dalloro, "Networking elements as a patch distribution platform for distributed automation and control domains," U.S. Patent PCT/US2012/043 084, Siemens Corporate Research, June, 19th, 2012.

[45] A. Wander, N. Gura, H. Eberle, V. Gupta, and S. Shantz, "Energy analysis of public-key cryptography for wireless sensor networks," in *Pervasive Computing and Communications, 2005. PerCom 2005. Third IEEE International Conference on.* IEEE, 2005, pp. 324–328.

[46] N. Gura, A. Patel, A. Wander, H. Eberle, and S. Shantz, "Comparing elliptic curve cryptography and rsa on 8-bit cpus," *Cryptographic Hardware and Embedded Systems-CHES 2004*, pp. 925–943, 2004.

[47] H. Rifa-Pous and J. Herrera-Joancomartí, "Computational and energy costs of cryptographic algorithms on handheld devices," *Future Internet*, vol. 3, no. 1, pp. 31–48, 2011.

[48] Wikipedia. (2012) ipaq personal digital assistant. [Online]. Available: http://en.wikipedia.org/wiki/iPAQ

[49] Crossbow. (2012) Mica2 mote data sheet.

[50] X. Du, M. Guizani, Y. Xiao, and H. Chen, "Two tier secure routing protocol for heterogeneous sensor networks," *Wireless Communications, IEEE Transactions on*, vol. 6, no. 9, pp. 3395–3401, 2007.

[51] A. Aziz and W. Diffie, "Privacy and authentication for wireless local area networks," *Personal Communications, IEEE*, vol. 1, no. 1, pp. 25–31, 1994.

[52] H. Wang, B. Sheng, C. Tan, and Q. Li, "Comparing symmetric-key and public-key based security schemes in sensor networks: A case study of user access control," in *Distributed Computing Systems, 2008. ICDCS'08. The 28th International Conference on.* IEEE, 2008, pp. 11–18.

[53] G. Gaubatz, J. Kaps, E. Ozturk, and B. Sunar, "State of the art in ultra-low power public key cryptography for wireless sensor networks," in *Pervasive Computing and Communications Workshops, 2005. PerCom 2005 Workshops. Third IEEE International Conference on.* IEEE, 2005, pp. 146–150.

[54] E. Blaß and M. Zitterbart, "Towards acceptable public-key encryption in sensor networks," in *ACM 2nd International Workshop on Ubiquitous Computing.* INSTICC Press Miami, USA, 2005, pp. 88–93.

[55] G. Gaubatz, J. Kaps, and B. Sunar, "Public key cryptography in sensor networks, revisited," in *Proceedings of the First European conference on Security in Ad-hoc and Sensor Networks.* Springer-Verlag, 2004, pp. 2–18.

[56] ——, "Public key cryptography in sensor networks, revisited," *Security in Ad-hoc and Sensor Networks*, pp. 2–18, 2005.

[57] D. Malan, M. Welsh, and M. Smith, "A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography," in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on.* IEEE, 2004, pp. 71–80.

[58] K. Piotrowski, P. Langendoerfer, and S. Peter, "How public key cryptography influences wireless sensor node lifetime," in *Proceedings of the fourth ACM workshop on Security of ad hoc and sensor networks.* ACM, 2006, pp. 169–176.

[59] L. Batina, N. Mentens, K. Sakiyama, B. Preneel, and I. Verbauwhede, "Low-cost elliptic curve cryptography for wireless sensor networks," *Security and Privacy in Ad-Hoc and Sensor Networks*, pp. 6–17, 2006.

[60] P. Trakadas, T. Zahariadis, H. Leligou, S. Voliotis, and K. Papadopoulos, "Analyzing energy and time overhead of security mechanisms in wireless sensor networks," in *Systems, Signals and Image Processing, 2008. IWSSIP 2008. 15th International Conference on.* IEEE, 2008, pp. 137–140.

[61] N. Potlapally, S. Ravi, A. Raghunathan, and N. Jha, "Analyzing the energy consumption of security protocols," in *Proceedings of the 2003 international symposium on Low power electronics and design.* ACM, 2003, pp. 30–35.

[62] ——, "A study of the energy consumption characteristics of cryptographic algorithms and security protocols," *Mobile Computing, IEEE Transactions on*, vol. 5, no. 2, pp. 128–143, 2006.

[63] K. Malhotra, S. Gardner, and R. Patz, "Implementation of elliptic-curve cryptography on mobile healthcare devices," in *Networking, Sensing and Control, 2007 IEEE International Conference on.* IEEE, 2007, pp. 239–244.

[64] R. P. Institute. (2012) Sense - sensor network simulator and emulator. [Online]. Available: http://www.ita.cs.rpi.edu/sense/index.html

[65] I. S. Institute. (2013) ns-2 - network simulator and emulator. [Online]. Available: http://www.isi.edu/nsnam/ns/

[66] Q. Xue and A. Ganz, "Runtime security composition for sensor networks (securesense)," in *Vehicular Technology Conference, 2003. VTC 2003-Fall. 2003 IEEE 58th*, vol. 5. IEEE, 2003, pp. 2976–2980.

[67] H. Lee, Y. Choi, and H. Kim, "Implementation of tinyhash based on hash algorithm for sensor network," in *Proceedings of World Academy of Science, Engineering and Technology*, vol. 10, 2005, pp. 135–139.

[68] F. Rodríguez-Henríquez, C. López-Peza, M. León-Chávez, and P. Puebla, "Comparative performance analysis of public-key cryptographic operations in the wtls handshake protocol," in *Electrical and Electronics Engineering, 2004.(ICEEE). 1st International Conference on.* IEEE, 2004, pp. 124–129.

[69] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *Communications Surveys and Tutorials, IEEE*, vol. 10, no. 3, pp. 6–28, 2008.

[70] X. Chen, K. Makki, K. Yen, and N. Pissinou, "Sensor network security: a survey," *Communications Surveys and Tutorials, IEEE*, vol. 11, no. 2, pp. 52–73, 2009.

[71] A. Fuchsberger, "Intrusion detection systems and intrusion prevention systems," *Information Security Technical Report*, vol. 10, no. 3, pp. 134–139, 2005.

[72] M. Ngadi, A. Abdullah, S. Mandala, *et al.*, "A survey on manet intrusion detection," *International Journal of Computer Science and Security*, vol. 2, no. 1, pp. 1–11, 2008.

[73] Y. Zhang, W. Lee, and Y. Huang, "Intrusion detection techniques for mobile wireless networks," *Wireless Networks*, vol. 9, no. 5, pp. 545–556, 2003.

[74] A. Patcha and J. Park, "An overview of anomaly detection techniques: Existing solutions and latest technological trends," *Computer Networks*, vol. 51, no. 12, pp. 3448–3470, 2007.

[75] T. Sobh, "Wired and wireless intrusion detection system: Classifications, good characteristics and state-of-the-art," *Journal of Computer Standards and Interfaces*, vol. 28, no. 6, pp. 670–694, 2006.

[76] T. Anantvalee and J. Wu, "A survey on intrusion detection in mobile ad hoc networks," *Wireless Network Security*, pp. 159–180, 2007.

[77] P. Albers, O. Camp, J. Percher, B. Jouga, L. Mé, and R. Puttini, "Security in ad hoc networks: A general intrusion detection architecture enhancing trust based approaches," in *Proceedings of the First International Workshop on Wireless Information Systems (WIS-2002)*, 2002, pp. 1–12.

[78] P. Michiardi and R. Molva, "Core: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proceedings of the IFIP TC6/TC11 Sixth Joint Working Conference on Communications and Multimedia Security: Advanced Communications and Multimedia Security*. Kluwer, BV, 2002, pp. 107–121.

[79] P. Garcia-Teodoro, J. Diaz-Verdejo, G. Maciá-Fernández, and E. Vázquez, "Anomaly-based network intrusion detection: Techniques, systems and challenges," *Computers and Security*, vol. 28, no. 1, pp. 18–28, 2009.

[80] Y. Huang and W. Lee, "A cooperative intrusion detection system for ad hoc networks," in *Workshop on Security of ad hoc and Sensor Networks: Proceedings of the 1 st ACM workshop on Security of ad hoc and sensor networks*, vol. 2003, 2003, pp. 135–147.

[81] A. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. Wong, "Decentralized intrusion detection in wireless sensor networks," in *Proceedings of the 1st ACM international workshop on Quality of service and security in wireless and mobile networks*. ACM, 2005, pp. 16–23.

[82] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," in *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*. IEEE, 2003, pp. 368–373.

[83] B. Sun, L. Osborne, Y. Xiao, and S. Guizani, "Intrusion detection techniques in mobile ad hoc and wireless sensor networks," *Wireless Communications, IEEE*, vol. 14, no. 5, pp. 56–63, 2007.

[84] A. Farooqi and F. Khan, "Intrusion detection systems for wireless sensor networks: A survey," *Communication and networking*, pp. 234–241, 2009.

[85] C. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Computers and Security*, vol. 29, no. 1, pp. 124–140, 2010.

[86] H. Elshoush and I. Osman, "Alert correlation in collaborative intelligent intrusion detection systemsa survey," *Applied Soft Computing*, vol. 11, no. 7, pp. 4349–4365, 2011.

[87] C. Vincent Zhou, C. Leckie, and S. Karunasekera, "Decentralized multi-dimensional alert correlation for collaborative intrusion detection," *Journal of Network and Computer Applications*, vol. 32, no. 5, pp. 1106–1123, 2009.

[88] C. Modi, D. Patel, B. Borisaniya, H. Patel, A. Patel, and M. Rajarajan, "A survey of intrusion detection techniques in cloud," *Journal of Network and Computer Applications*, vol. 36, pp. 42–57, 2013.

[89] K. Garcia, R. Monroy, L. Trejo, C. Mex-Perera, and E. Aguirre, "Analyzing log files for post-mortem intrusion detection," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 42, no. 6, pp. 1690–1704, 2012.

[90] T. Cheng, Y. Lin, Y. Lai, and P. Lin, "Evasion techniques: Sneaking through your intrusion detection/prevention systems," *Communications Surveys and Tutorials, IEEE*, vol. 14, no. 4, pp. 1011–1020, 2012.

[91] G. Keung, B. Li, and Q. Zhang, "The intrusion detection in mobile sensor network," *IEEE/ACM Transactions on Networking (TON)*, vol. 20, no. 4, pp. 1152–1161, 2012.

[92] Y. Wang, W. Fu, and D. Agrawal, "Gaussian versus uniform distribution for intrusion detection in wireless sensor networks," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 2, pp. 342–355, 2012.

[93] E. Shakshuki, N. Kang, and T. Sheltami, "Eaack– a secure intrusion detection system for manets," *IEEE Transactions on Industrial Electronics, vol. 60, num. 3, pp. 108*, vol. 60, no. 3, pp. 1089–1098, 2013.

[94] Y. Zhang and W. Lee, "Intrusion detection in wireless ad-hoc networks," in *Proceedings of the 6th annual international conference on Mobile computing and networking.* ACM, 2000, pp. 275–283.

[95] O. Kachirski and R. Guha, "Intrusion detection using mobile agents in wireless ad hoc networks," in *Knowledge Media Networking, 2002. Proceedings. IEEE Workshop on.* IEEE, 2002, pp. 153–158.

[96] D. Sterne, P. Balasubramanyam, D. Carman, B. Wilson, R. Talpade, C. Ko, R. Balupari, C. Tseng, and T. Bowen, "A general cooperative intrusion detection architecture for manets," in *Information Assurance, 2005. Proceedings. Third IEEE International Workshop on.* IEEE, 2005, pp. 57–70.

[97] R. Puttini, M. Hanashiro, F. Miziara, R. de Sousa, L. García-Villalba, and C. Barenco, "On the anomaly intrusion-detection in mobile ad hoc network environments," in *Personal Wireless Communications.* Springer, 2006, pp. 182–193.

[98] K. Nadkarni and A. Mishra, "Intrusion detection in manets-the second wall of defense," in *Industrial Electronics Society, 2003. IECON'03. The 29th Annual Conference of the IEEE*, vol. 2. IEEE, 2003, pp. 1235–1238.

[99] S. Buchegger and J. Le Boudec, "Performance analysis of the confidant protocol," in *Proceedings of the 3rd ACM international symposium on Mobile ad hoc networking and computing.* ACM, 2002, pp. 226–236.

[100] B. Sun, K. Wu, and U. Pooch, "Zone-based intrusion detection for mobile ad hoc networks," *Int. Journal of Ad Hoc and Sensor Wireless Networks*, vol. 2, no. 3, 2003.

[101] A. Patcha and J. Park, "A game theoretic approach to modeling intrusion detection in mobile ad hoc networks," in *Information Assurance Workshop, 2004. Proceedings from the Fifth Annual IEEE SMC.* IEEE, 2004, pp. 280–284.

[102] ——, "A game theoretic formulation for intrusion detection in mobile ad hoc networks," *International Journal of Network Security*, vol. 2, no. 2, pp. 131–137, 2006.

[103] S. Sen and J. Clark, "Evolutionary computation techniques for intrusion detection in mobile ad hoc networks," *Computer Networks*, vol. 55, no. 15, pp. 3441–3457, 2011.

[104] S. Marti, T. Giuli, K. Lai, M. Baker, *et al.*, "Mitigating routing misbehavior in mobile ad hoc networks," in *International Conference on Mobile Computing and Networking: Proceedings of the 6 th annual international conference on Mobile computing and networking*, vol. 6, no. 11, 2000, pp. 255–265.

[105] F. Wai, Y. Aye, and N. James, "Intrusion detection in wireless ad-hoc networks," *Term Paper, School of Computing, National University of Singapore*, 2003.

[106] M. Jahnke, G. Klein, A. Wenzel, N. Aschenbruck, E. Gerhards-Padilla, P. Ebinger, S. Karsch, and J. Haag, "Mite–manet intrusion detection for tactical environments," in *Proc. of the NATO/RTO IST-076 Research Symposium on Information Assurance for Emerging and Future Military Systems, Ljubljana, Slovenia*, 2008.

[107] M. Wei and K. Kim, "Intrusion detection scheme using traffic prediction for wireless industrial networks," *Communications and Networks, Journal of*, vol. 14, no. 3, pp. 310–318, 2012.

[108] A. Mishra, K. Nadkarni, and A. Patcha, "Intrusion detection in wireless ad hoc networks," *Wireless Communications, IEEE*, vol. 11, no. 1, pp. 48–60, 2004.

[109] S. Şen and J. Clark, "Intrusion detection in mobile ad hoc networks," *Guide to Wireless Ad Hoc Networks*, pp. 427–454, 2009.

[110] R. Roman, J. Zhou, and J. Lopez, "Applying intrusion detection systems to wireless sensor networks," in *Proceedings of IEEE Consumer Communications and Networking Conference (CCNC06)*, 2006, pp. 640–644.

[111] Crossbow. (2012) Micaz datasheet.

[112] Atmel. (2012) Atmel atmega128l microcontroller datasheet. [Online]. Available: http://www.atmel.com/Images/doc2467.pdf

[113] Intel. (2012) Intel notebook processors. [Online]. Available: http://www.intel.com/support/processors/mobile/pm/sb/cs-007967.htm

[114] Digi. (2012) Xbee wifi module datasheet. [Online]. Available: http://www.digi.com/pdf/ds_xbeewifi.pdf

[115] A. Strikos, "A full approach for intrusion detection in wireless sensor networks," *School of Information and Communication Technology, KTH*, 2007.

[116] K. Ioannis, T. Dimitriou, and F. Freiling, "Towards intrusion detection in wireless sensor networks," in *Proc. of the 13th European Wireless Conference*. Citeseer, 2007.

[117] I. Krontiris, Z. Benenson, T. Giannetsos, F. Freiling, and T. Dimitriou, "Cooperative intrusion detection in wireless sensor networks," *Wireless Sensor Networks*, pp. 263–278, 2009.

[118] E. Ngai, J. Liu, and M. Lyu, "On the intruder detection for sinkhole attack in wireless sensor networks," in *Communications, 2006. ICC'06. IEEE International Conference on*, vol. 8. IEEE, 2006, pp. 3383–3389.

[119] S. Doumit and D. Agrawal, "Self-organized criticality and stochastic learning based intrusion detection system for wireless sensor networks," in *Military Communications Conference, 2003. MILCOM'03. 2003 IEEE*, vol. 1. IEEE, 2003, pp. 609–614.

[120] I. Onat and A. Miri, "A real-time node-based traffic anomaly detection algorithm for wireless sensor networks," in *Systems Communications, 2005. Proceedings.* IEEE, 2005, pp. 422–427.

[121] S. Rajasegarar, C. Leckie, and M. Palaniswami, "Anomaly detection in wireless sensor networks," *Wireless Communications, IEEE*, vol. 15, no. 4, pp. 34–40, 2008.

[122] S. Rajasegarar, C. Leckie, M. Palaniswami, and J. Bezdek, "Quarter sphere based distributed anomaly detection in wireless sensor networks," in *Communications, 2007. ICC'07. IEEE International Conference on.* IEEE, 2007, pp. 3864–3869.

[123] V. Bhuse and A. Gupta, "Anomaly intrusion detection in wireless sensor networks," *Journal of High Speed Networks*, vol. 15, no. 1, pp. 33–51, 2006.

[124] I. Onat and A. Miri, "An intrusion detection system for wireless sensor networks," in *Wireless And Mobile Computing, Networking And Communications, 2005.(WiMob'2005), IEEE International Conference on*, vol. 3. IEEE, 2005, pp. 253–259.

[125] C. Wang, T. Feng, J. Kim, G. Wang, and W. Zhang, "Catching packet droppers and modifiers in wireless sensor networks," *Parallel and Distributed Systems, IEEE Transactions on*, vol. 23, no. 5, pp. 835–843, 2012.

[126] F. Bao, R. Chen, M. Chang, and J. Cho, "Hierarchical trust management for wireless sensor networks and its applications to trust-based routing and intrusion detection," *Network and Service Management, IEEE Transactions on*, vol. 9, no. 2, pp. 169–183, 2012.

[127] O. Adaobi, E. Igbesoko, and M. Ghassemian, "Evaluation of security problems and intrusion detection systems for routing attacks in wireless self-organised networks," in *New Technologies, Mobility and Security (NTMS), 2012 5th International Conference on*. IEEE, 2012, pp. 1–5.

[128] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "An application-specific protocol architecture for wireless microsensor networks," *IEEE Transactions On Wireless Communications*, vol. 1, no. 4, 2002.

[129] I. Krontiris, T. Dimitriou, T. Giannetsos, and M. Mpasoukos, "Intrusion detection of sinkhole attacks in wireless sensor networks," *Algorithmic Aspects of Wireless Sensor Networks*, pp. 150–161, 2008.

[130] J. Brown and X. Du, "Detection of selective forwarding attacks in heterogeneous sensor networks," in *Communications, 2008. ICC'08. IEEE International Conference on*. IEEE, 2008, pp. 1583–1587.

[131] H. Yang, J. Shu, X. Meng, and S. Lu, "Scan: self-organized network-layer security in mobile ad hoc networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 261–273, 2006.

[132] A. Wood and J. Stankovic, "Denial of service in sensor networks," *IEEE Computer*, pp. 54–62, 2002.

[133] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and countermeasures," *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.

[134] D. R. Raymond and S. F. Midkiff, "Denial-of-service in wireless sensor networks: Attacks and defenses," *Pervasive Computing, IEEE*, vol. 7, no. 1, pp. 74–81, 2008.

[135] Y. Hu, A. Perrig, and D. Johnson, "Rushing attacks and defense in wireless ad hoc network routing protocols," in *Proceedings of the 2nd ACM workshop on Wireless security*. ACM, 2003, pp. 30–40.

[136] J. Douceur, "The sybil attack," Peer-To-Peer Systems: First International Workshop. Cambridge, Ma,: Springer, 2002, p. 251.

[137] Y. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications. IEEE Societies*, vol. 3. IEEE, 2003, pp. 1976–1986.

[138] E. Cayirci and C. Rong, *Journal of Security in Wireless Ad Hoc and Sensor Networks*. Wiley, 2009.

[139] H. Yang, H. Luo, F. Ye, S. Lu, and L. Zhang, "Security in mobile ad hoc networks: challenges and solutions," *IEEE Wireless Communications*, vol. 11, no. 1, pp. 38–47, 2004.

[140] F. Hu and N. Sharma, "Security considerations in ad hoc sensor networks," *Ad Hoc Networks*, vol. 3, no. 1, pp. 69–89, 2005.

[141] F. Stajano and R. Anderson, "The resurrecting duckling: security issues for ubiquitous computing," *IEEE Journal of Computer*, vol. 35, no. 4, pp. 22–26, 2002.

[142] B. Parno, A. Perrig, and V. Gligor, "Distributed detection of node replication attacks in sensor networks," in *Security and Privacy, 2005 IEEE Symposium on*. IEEE, 2005, pp. 49–63.

[143] R. Muraleedharan and L. Osadciw, "Cross layer denial of service attacks in wireless sensor network using swarm intelligence," in *Information Sciences and Systems, 2006 40th Annual Conference on*. IEEE, 2006, pp. 1653–1658.

[144] Y. Zhang, W. Liu, W. Lou, and Y. Fang, "Location-based compromise-tolerant security mechanisms for wireless sensor networks," *Selected Areas in Communications, IEEE Journal on*, vol. 24, no. 2, pp. 247–260, 2006.

[145] J. Deng, R. Han, and S. Mishra, "A performance evaluation of intrusion-tolerant routing in wireless sensor networks," *Lecture Notes in Computer Science*, vol. 2634, pp. 349–364, 2003.

[146] R. Di Pietro, L. Mancini, Y. Law, S. Etalle, and P. Havinga, "Lkhw: A directed diffusion-based secure multicast scheme for wireless sensor networks," in *Parallel Processing Workshops, 2003. Proceedings. 2003 International Conference on*. IEEE, 2003, pp. 397–406.

[147] M. Tubaishat, J. Yin, B. Panja, and S. Madria, "A secure hierarchical model for sensor network," *ACM SIGMOD Record*, vol. 33, no. 1, pp. 7–13, 2004.

[148] B. Yu and B. Xiao, "Detecting selective forwarding attacks in wireless sensor networks," in *Parallel and Distributed Processing Symposium, 2006. IPDPS 2006. 20th International*. IEEE, 2006, pp. 8–pp.

[149] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly-resilient, energy-efficient multipath routing in wireless sensor networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 5, no. 4, pp. 11–25, 2001.

[150] G. Wang, W. Zhang, G. Cao, and T. La Porta, "On supporting distributed collaboration in sensor networks," in *Military Communications Conference, 2003. MILCOM'03. 2003 IEEE*, vol. 2. IEEE, 2003, pp. 752–757.

[151] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: analysis and defenses," in *Proceedings of the 3rd international symposium on Information processing in sensor networks*. ACM, 2004, pp. 259–268.

[152] L. Hu and D. Evans, "Using directional antennas to prevent wormhole attacks," in *Network and Distributed System Security Symposium (NDSS)*. San Diego, 2004.

[153] W. Wang and B. Bhargava, "Visualization of wormholes in sensor networks," in *Proceedings of the 3rd ACM workshop on Wireless security*. ACM, 2004, pp. 51–60.

[154] M. Khalid, Y. Wang, I. Butun, H. Kim, I. Ra, and R. Sankar, "Coherence time-based cooperative mac protocol for wireless ad hoc networks," *EURASIP Journal on Wireless Communications and Networking*, vol. 2011, no. 1, pp. 1–12, 2011.

[155] I. Butun, S. Birla, X. Le, S. Lee, and R. Sankar, "Performance evaluation of quick-start in low latency networks," in *Consumer Communications and Networking Conference (CCNC), 2010 7th IEEE*. IEEE, 2010, pp. 1–3.

[156] I. Butun, A. Cagatay Talay, D. Turgay Altilar, M. Khalid, and R. Sankar, "Impact of mobility prediction on the performance of cognitive radio networks," in *Wireless Telecommunications Symposium (WTS), 2010.* IEEE, 2010, pp. 1–5.

[157] Y. Xu, I. Butun, R. Sankar, N. Sapankevych, and J. Crain, "Comparison of routing and network coding in undirected network group communications," in *Southeastcon, 2012 Proceedings of IEEE.* IEEE, 2012, pp. 1–6.

[158] Y. Wang, I. Butun, R. Sankar, and S. Morgera, "Adaptive rate transmission with opportunistic scheduling in wireless networks," in *Consumer Communications and Networking Conference (CCNC), 2012 IEEE.* IEEE, 2012, pp. 445–448.

[159] Y. Wang, I. Butun, R. Sankar, and I. Ra, "Delay considerations with two-hop opportunistic relays in wireless networks," in *International Conference on Smart Media and Applications (SMA)*, 2012.

[160] I. Butun and M. Demirer, "A blind digital signature scheme using elliptic curve digital signature algorithm," *The Turkish Journal of Electrical Engineering and Computer Sciences*, vol. 1, no. 1, pp. 1–2, 2013.

APPENDICES

Appendix A

A.1 Security Vocabulary

Access control: Ensures that all accesses to objects (information resources) are authorized by regulating different privileged operations.

Attack: A specific formulation or execution of a plan to carry out a threat.

Audit: It is the process of gathering data about activity in the system and analyzes it to discover security violations or diagnose their cause.

Authentication: It establishes a relation between a user and some identity (password, secret key, token, etc.).

Authorization: Establishing a relation between a user and a set of privileges (access rights, allowed operations (read-write, read-only, etc.)).

Availability: The network should always be able to answer any authorized request in its life time before the request expires.

Confidentiality: Only authorized parties should be able to access the data.

Integrity: If an authorized user receives data, this data should be correct and valid; it shouldn't be changed by unauthorized parties.

Intrusion: A set of actions that are planned to compromise the security goals (integrity, confidentiality, and availability) of a computer system.

Non-repudiation: Neither the sender, nor the receiver can deny the transaction of the message.

Penetration: The ability to get unauthorized access to a computer system as a result of a successful attack.

Risk: Accidental exposure of information, or violation of operation integrity due to the vulnerabilities in the system.

Vulnerability: A flaw in the system that exposes its information to accidental disclosure.

Appendix B

B.1 Attacks towards the Wireless Sensor Networks

In the literature, there is a variety of classifications for attacks towards the Wireless Sensor Networks (WSNs) [80, 132–134]. Following, a brief summary of these classifications is provided:

- *According to Source of the Attack:* Internal (insider)/external (outsider) attacks. Intrusion prevention mechanisms can catch external attacks but not the internal attacks. The only way of reacting against internal attacks is using the Intrusion Detection Systems (IDSs). After an intrusion is detected then a prevention mechanism would be issued to minimize the adverse effects.
- *According to Participants:* Host-based/network-based attacks.
- *According to Activity of the Attacker:* Passive/active attacks.
- *According to the Targeted OSI Protocol Layer:* Security attacks can also be categorized based on the OSI protocol layers that are being targeted for node compromise:
  - Physical layer attacks: radio interference, jamming, DoS.
  - Data link (MAC) layer attacks: sleep deprivation torture (denial of sleep).
  - Network layer attacks: sinkhole, wormhole, blackhole, selective forwarding, Sybil, HELLO flooding.
  - Transport layer attacks: memory exhaustion attack.
  - Application layer attacks: information gathering attack.
- *According to the Techniques Used to Perform the Attack:*
  - Cash poisoning: Information stored in routing tables are modified, deleted or injected with bogus data.
  - Fabricated route messages: Route messages (e.g. request, reply, error, etc.) that contain malicious data are injected into the network.
    - ∗ False resource route: False route information is advertised throughout the network (e.g. setting the route hope count to minimum regardless of the destination).

        * Maximum sequence: Modifying the sequence field in control messages to exceed the maximal allowed value, which would invalidate all legitimate messages although they normally have sequence time in the allowed ranges.

    - Flooding: Delivering unusual large amount of data or control packets to clog the network.
    - Packet dropping: A node drops data packets (conditionally or randomly) that it was supposed to forward.
    - Rushing: Uses a weakness that some of the routing protocols possess; whichever routing message arrives first to the recipient is accepted as the valid route and the others are rejected (first come, first served). The attacker exploits this vulnerability by broadcasting malicious control messages quickly to block legitimate control messages that arrive later on [135].
    - Spoofing: Injecting data or control packets with modified source addresses to imitate as if they were sourced by legitimate users.
    - Sybil: A single node presents multiple identities to other nodes of the network. This causes confusion in the network; nodes receive contradicting routing paths that are passing through the attacker [136].
    - Wormhole: A tunnel is created (by out of the band, high transmission connection) between two nodes that can be utilized to secretly transmit packets, which would cause confusion and/or delusion in the network [137].

Among these, we will use both "According to source of the attack" and "According to the targeted OSI protocol layer" classifications as shown in Figure B.1. Following subsections include descriptions of each item in the Figure B.1.

B.1.1 Passive Attacks

Passive attacks are performed in a way that it cannot be sensed by any means. This is because of the fact that the adversaries do not make any radio emissions. Since wireless links are easier to

Appendix B (Continued)

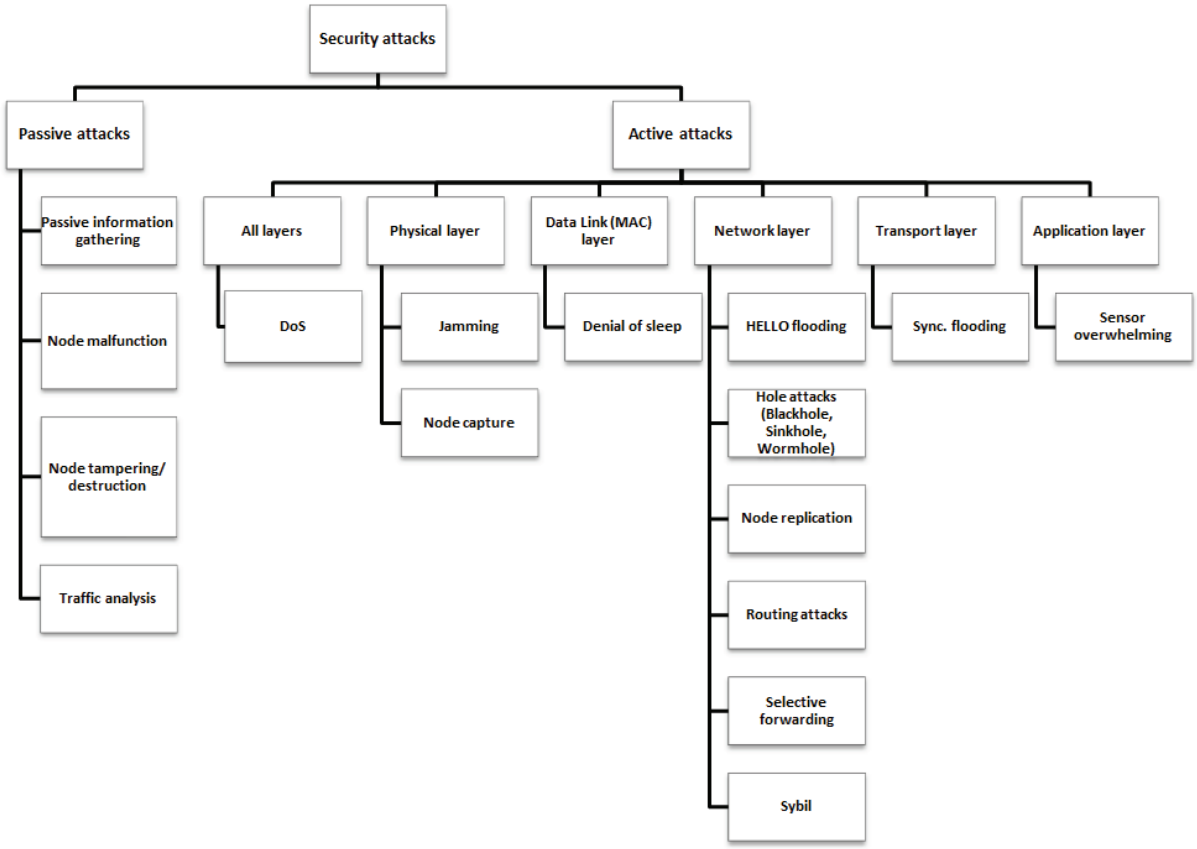

Figure B.1 Security attacks towards the WSNs - OSI layered description

tap, wireless networks are more susceptible to passive attacks, such as eavesdropping, which can be performed easily listening to the wireless communication amongst sensor nodes in the WSN without capturing any of them. Passive attacks are mainly against data confidentiality.

In passive attacks, attackers are typically camouflaged, i.e. hidden, and tap the communication lines to collect data. Passive attacks can be grouped into eavesdropping, node malfunctioning, node tampering/destruction and traffic analysis types (see Figure B.1).

Passive Information Gathering (Eavesdropping): Eavesdropping is also known as "Passive information gathering". Classified data can be eavesdropped by tapping communication lines, and wireless links are easier to tap. Therefore, wireless networks are more susceptible to passive attacks.

Since WSNs use short range communications, attacker must be in proximity in order to gather useful information.

WSNs are a little more secure against tapping compared to other longer range wireless technologies, because signals are sent over shorter distances.

Node Malfunctioning: This may happen due to many different factors from faulty sensors or energy depletion due to sensor overwhelming or other DoS attacks.

Node Tampering/Destruction: Physically destruction (with the usage of electrical surge, physical force or ammunition) or tampering (changing the wiring of the electronic board, memory, etc.) of the nodes by any means.

Traffic Analysis: As well as the content of data packets, the traffic pattern may also be very valuable for adversaries. Important information about the networking topology can be derived by analyzing traffic patterns. In WSNs, the nodes closer to the base station, i.e. the sink, make more transmissions than the other nodes because they relay more packets than the nodes farther from the base station. Similarly, clustering is an important tool for scalability in WSNs and cluster heads are busier than the other nodes in the network. Detection of the base station, the nodes close to it or cluster heads may be very useful for adversaries because a denial-of-service attack against these nodes or eavesdropping the packets destined for them may have a greater impact. By analyzing the traffic, this kind of valuable information can be derived.

Moreover, traffic patterns can pertain to other confidential information such as actions and intentions. In tactical communications, silence may indicate preparation for an attack, a tactical move or infiltration. Similarly, a sudden increase in the traffic rate may indicate the start of a deliberate attack or raid.

B.1.2 Active Attacks

In active attacks, malicious acts are carried out not only against data confidentiality but also data integrity. Active attacks can also aim for unauthorized access and usage of the resources or the disturbance of an opponent's communications. An active attacker makes a radio emission or action that can be sensed by the WSN elements [138]. An example is DoS attack in the physical and/or network layer that would cause network elements to drop data packets.

In active attacks, an adversary actually affects the operations in the attacked network. This effect may be the objective of the attack and can be detected. For example, the networking services may be degraded or terminated as a result of these attacks. Sometimes the adversary tries to stay undetected, aiming to gain unauthorized access to the system resources or threatening confidentiality and/or integrity of the content of the network. Active attacks that we are interested for WSNs grouped into two main groups, attacks towards for all layers and attacks towards network layer. Network layer attacks are divided into seven classes, as shown in Figure B.1.

B.1.2.1 Attacks Towards all Layers

Denial-of-Service (DoS): A denial-of-service (DoS) attack mainly targets the availability of network services. A DoS is defined as any event that diminishes a network's capacity to perform its expected function correctly or in a timely manner. A node is isolated from the rest of the network by blocking the incoming and outgoing packets.

In DoS attack, an adversary attempts to prevent legitimate and authorized users of services offered by the network from accessing those services. The classic way to achieve this is to flood packets to any centralized resource (access point) used in the network so that the resource is no longer available to the nodes in the network, resulting the network no longer operating what was designed for. This may lead to a failure in the delivery of guaranteed services to the end users.

DoS attack in the physical layer is called *jamming*. A malicious device can jam a wireless carrier by transmitting a signal at that frequency. The jamming signal contributes to the noise in the carrier and its strength is enough to reduce the signal-to-noise ratio below the level that the nodes using that channel need to receive data correctly. Jamming can be conducted continuously in a region, which thwarts all the nodes in that region from communication. Alternatively, jamming can be done temporarily with random time intervals, which can still very effectively hamper the transmissions.

The algorithms in the link layer, especially MAC schemes, present many exploitation opportunities for DoS attacks. For example, MAC layer DoS attacks may continuously jam a channel. More complex DoS attacks can be designed based on MAC layer addressing schemes.

In the case of network layer DoS attack, an attacker injects significant amount of packets into the network which causes congestion in the network traffic as well as deprivation of power resources

Table B.1 DoS attacks towards WSNs [140].

DoS attacks	**Meaning**
Radio interferences	Jamming of the radio transmission in the MAC or physical layer
Physical tampering	An attacker captures and compromises the sensor nodes
Denying channel	An attacker uses collision to damage the wireless channel and causes packets to be dropped
Black holes	A malicious node in the route sinks and drops messages that are routed through them
Misdirection	An attacker uses loop or detour to misdirect traffic
Flooding	A malicious node floods lots of messages to cause congestion and energy exhaustion
Anti-synchronization	An attacker forges timing control messages to disrupt the synchronization between two nodes
Critical attack	An attacker learns the critical resources such as cluster heads and attacks them.

throughout the network. Examples: "Routing table overflow attack: Creation of the routes to the non-existing nodes", "sleep deprivation attack: deprivation of the power supplies of a targeted node" [139].

Application layer protocols can also be exploited in DoS attacks. Protocols like node localization, time synchronization, data aggregation, association and fusion can be cheated or hindered. For example, a malicious node that impersonates a beacon node and gives false location information or cheats with regard to its transmission power, i.e. transmitting with less or more power than it is supposed to do, may hamper the node localization scheme. Since these kinds of attack diminish the related network service, they can also be categorized as DoS attacks.

All the possible DoS attacks against WSNs are summarized in Table B.1.

B.1.2.2 Attacks Towards Physical Layer

Jamming: It is a DoS attack at the physical layer. A malicious device can jam a signal by transmitting in the same frequency. The jamming signal contributes to the noise in the carrier and its strength is enough to reduce the signal-to-noise ratio below the level that the nodes using that channel need to receive data correctly. Jamming can be conducted continuously in a region, which thwarts all the nodes in that region from communication. Alternatively, jamming can be done temporarily with random time intervals, which can still very effectively hamper the transmissions.

Node Capture Attack: An adversary takes over the control of the sensor node by a physical attack, e.g. attaching cables to its circuit board and reading stored data as well as ongoing transmission in the WSN. Two problems arise in this case:

- Captured node can make arbitrary queries on behalf of the attacker (DoS attack against availability).
- Captured node can provide false data to the legitimate users (attack against integrity).

B.1.2.3 Attacks Towards Data Link (MAC) Layer

Sleep Deprivation Torture (Denial of sleep): Preventing a node from going to sleep leading to energy depletion from draining the battery. This can be from collision attacks or repeated handshaking (RTS/CTS). In this attack, a node is forced to deplete whole energy stored in its batteries [141].

B.1.2.4 Attacks Towards Network Layer

HELLO Flooding Attack: Attacker (has longer transmission range than normal nodes) broadcasts advertisement messages to whole network and convinces other nodes that it is located in their neighborhood.

Routing protocols broadcast "HELLO" message to inform of their presence to one-hop neighbors. A node receiving such a packet assumes that it is within the radio range of the sender which may not be true during this attack. A malicious node may flood "HELLO" packets with high enough transmission power to convince every node in the network that it is their neighbor. When the other nodes send their packets to the malicious node, those packets are not received by any node.

Many protocols require nodes to broadcast HELLO packets to announce themselves to their neighbors, and a node receiving such a packet may assume that it is within (normal) radio range of the sender. This assumption may be false: a laptop-class attacker broadcasting routing or other information with large enough transmission power could convince every node in the network that the adversary is its neighbor.

Appendix B (Continued)

"Flooding" is usually used to denote the epidemic-like propagation of a message to every node in the network over a multi-hop topology. In contrast, despite its name, the HELLO flood attack uses a single hop broadcast to transmit a message to a large number of receivers [133].

Hole Attacks

- *Blackhole Attack:* A malicious node may drop all the packets that it receives for forwarding. This attack is especially effective when the black hole node is also a sink hole. Such an attack combination may stop all the data traffic around the black hole. In some texts, this attack is also referred as "Selfishness".

- *Sinkhole Attack:* All the traffic of the network is directed to a single node but in this case it does not drop any packets. By this way, expects to remain un-detected by the IDS. Since the all traffic of the network passes through this particular node which literally "sinks" all the data it receives, the name is given to this attack.

  A malicious node can advertise by broadcasting to all the neighbor nodes that it is the best next hop for sending the packets to its destination. When a node becomes a sink hole, it becomes the hub for its vicinity and starts receiving all the packets which are dropped.

  A malicious node can be made very attractive to the surrounding nodes with respect to the routing algorithm. For example, very attractive routing advertisements can be broadcast and all the neighboring nodes can be convinced that the malicious node is the best next hop for sending the packets to the base station. When a node becomes a sink hole, it becomes the hub for its vicinity and starts receiving all the packets going to the base station. This creates many opportunities for follow-on attacks.

- *Wormhole Attack:* A tunnel (out of the band fast transmission path) is created between two nodes that can be utilized to transmit packets in a faster way. This way, two far parts of the network advertised as neighbors to attract the surrounding traffic.

  A malicious node can eavesdrop or receive data packets at a point and transfer them to another malicious node, which is at another part of the network, through an out-

of-band channel. The second malicious node then replays the packets. This makes all the nodes that can hear the transmissions by the second malicious node believe that the node that sent the packets to the first malicious node is their single-hop neighbor and they are receiving the packets directly from it. The packets that follow the normal route reach destination node, later than those conveyed through the wormhole and are therefore dropped because they do more hops - wormholes are typically established through faster channels. Wormholes are very difficult to detect and can impact on the performance of many network services such as time synchronization, localization and data fusion.

Node Replication Attack: An attacker intentionally puts replicas of a compromised node in many places in the network to incur inconsistency. Like the Sybil attack, the node replication attack also can enable attackers to subvert data aggregation, misbehavior detection, and voting protocols by injecting false data or suppressing legitimate data [142].

Routing Attacks

- *Network Partitioning:* A full connected network is portioned to sub-networks in which the nodes in different sub-networks cannot communicate each other although they are connected.

- *Routing Loop:* A routing loop is introduced in a route path. It is created by spoofing routing updates. Suppose an adversary can determine that node A and node B are within radio range of each other. An adversary can send a forged routing update to node B with a spoofed source address indicating it came from node A. Node B will then mark node A as its parent and rebroadcast the routing update. Node A will then hear the routing update from node B and mark B as its parent. Messages sent to either A or B will be forever forwarded in a loop between the two of them. This leads to energy depletion and eventual node/network failure [133].

- *Spoofed, Altered or Replayed Routing Information:* Routing information exchanged among nodes can be altered by malicious nodes to have a detrimental effect on the routing scheme.

Selective Forwarding Attack: It is a special kind of black hole attack, in which malicious node acts more cleverly and does not drop every packet it receives but the ones it selects. By this way, attacker expects to remain un-detected by the IDS.

Similar to sinkhole attacks, a malicious node subverts the routing protocol by making itself part of many routes but instead of dropping of all packets selectively drop some packets while forwarding others in order to avoid detection.

Forwarding packets is a major responsibility of a routing node. However, a malicious node intentionally may drop any packet and forward other ones.

Multi-hop networks are often based on the assumption that participating nodes will faithfully forward the messages they received. In a selective forwarding attack, malicious nodes may refuse to forward certain messages and simply drop them, ensuring that they are not propagated any further.

A simple form of this attack is when a malicious node behaves like a black hole and refuses to forward every packet it receives. However, such an attacker has the following risk: Neighboring nodes will conclude that it has failed and they may decide to seek another route. A more subtle form of this attack is when an adversary selectively forwards packets. An adversary interested in suppressing or modifying packets originating from a select few nodes can reliably forward the remaining traffic and limit suspicion of her wrongdoing.

Sybil Attack: A single node presents multiple identities to other nodes of the network. This causes confusion in the network; nodes receive contradicting routing paths that are passing through the attacker. This reduces the effectiveness of fault-tolerance schemes and poses a significant threat to geographic routing protocols. Apart from these services it may also affect the performance of other schemes such as misbehavior detection, voting-based algorithms, data aggregation and fusion and distributed storage.

B.1.2.5 Attacks Towards Transport Layer

Synchronization Flooding: An attacker sends multiple connection requests without ever completing the connection, thus overwhelming the buffer.

B.1.2.6 Attacks Towards Application Layer

Sensor Overwhelming: Attacking or altering sensitivity of the sensor measurements. Target sensors with spurious interference or completely overwhelm and inundate with false stimuli.

Appendix C

C.1 Solutions to Defend Against Various Attacks Towards the WSNs

Routing protocols can be designed such that an adversary cannot compromise nodes/messages or make the routing scheme dysfunction. This is the most effective approach with respect to the cost of the security scheme and effectiveness in defense of WSNs against the threats. Therefore, most of the techniques fall into this category. Preventive approaches are designed to counter known threats and may not be effective against new threats. Detection schemes for misbehaving or malfunctioning nodes can be designed in a more generic fashion. On the other hand, they can be more costly than preventive approaches. Finally, routing can be designed such that it still delivers the data packets to the destination when there is an attack. Such resilient techniques are also costly.

Following subsections provide solutions (strategies and techniques) to defend against various attacks towards the WSNs:

C.1.1 Solutions to Defend Against DoS Attacks

In [143], authors propose a cross-layer security mechanism, namely "Swarm Intelligence", to detect DoS attacks. They also provide countermeasures to mitigate this kind of attack.

In Table C.1, some of the solutions to defend WSNs against DoS attacks are summarized.

C.1.2 Solutions to Defend Against HELLO Flooding Attacks

One possible solution to this problem is provided in [133]: Force every node to authenticate each of its neighbors with an identity verification protocol using a trusted base station. If the protocol sends messages in both directions over the link between the nodes, HELLO floods are prevented when the adversary only has a powerful transmitter because the protocol verifies the bi-directionality of the link.

C.1.3 Solutions to Defend Against Node Replication Attack

Conventional methods to detect a node replication attack usually include centralized computing based on node locations or the number of simultaneous connections, which is vulnerable to the single-point failure. Distributed detection of the node replication attack was proposed in [142], where each node is assumed to know its location, and it is required to send its location to a set of

Table C.1 Solutions to defend WSNs against DoS attacks [13].

DoS attack	Defense strategy
Radio interference	Usage of spread-spectrum communication
Physical tampering	Usage of tamper-resistant nodes
Denying channel	Usage of error correction codes
Black holes	Usage of multiple routing paths
Misdirection	Usage of source authorization
Flooding	Limiting the total number of connections

witness nodes. If a witness node finds a contradiction in the location claims of a suspected node identity, this suspected node identity must be replicated many times. Asymmetric key technology is used here to guarantee the authenticity of location claims. A similar approach is discussed in [144]: Each node has a private key corresponding to its location, and the location based key can be used to detect node replicas.

C.1.4 Solutions to Defend Against Passive Information Gathering (Eavesdropping) Attacks

Link layer encryption would prevent outsider attacks such as eavesdropping, and some of the solutions are provided in [133, 145–147].

C.1.5 Solutions to Defend Against Selective Forwarding attacks

There are two approaches to defending against selective forwarding:

- Detecting the nodes that selectively forwarding.
- Developing routing schemes that are more resilient and can deliver packets even when there is a selective forwarding attack.

One approach to detecting the nodes that selectively forward is based on acknowledgements [148]. Every intermediate node that forwards a packet waits for an acknowledgement from the next hop. If the next hop node does not return the same number of acknowledgements as the number of packets sent, the node generates an alarm about the next hop node. However, compromised nodes can also generate acknowledgements for the packets that they dropped, which make this scheme fail.

Multipath routing can be an effective way to mitigate selective forwarding and black hole attacks [133]. This requires at least link-disjoint paths, where two paths may share some nodes but no link. Of course, node-disjoint paths, where two paths do not have any node in common, are better and reduce the risk of selective forwarding attack compared to link-disjoint paths. However, disjoint paths are not always available, and when paths are not disjoint, if the selectively forwarding node is the node common to all the paths, then the attack can become as effective as in single-path routing.

Braided paths [149] may have nodes in common, but have no links in common (i.e., no two consecutive nodes in common). The use of multiple braided paths may provide probabilistic protection against selective forwarding and use only localized information.

In [130], authors describe an efficient scheme for reporting packet drops. They also present an effective scheme, namely "Wald's Sequential Probability Ratio Test", for detecting the selective forwarding attack in a heterogeneous sensor network. According to presented simulation results, proposed scheme achieves high detection ratio and low false alarm rate.

Wang et al. [150] proposed a failure detection framework to detect the selective forwarding attack. The observation is that for a routing node, the number of packets it forwards must be equal to the number of packets it receives. In their framework, each sensor node can work under a promiscuous mode so that it can overhear the transmission of neighboring nodes. If a neighbor of a suspected node finds that the number of packets that the suspected node fails to forward exceeds a certain threshold, the neighbor can collaborate with other neighbors of the suspected node, and the opinions from the neighbors of the suspected node are collected to form a decision about the suspected node.

C.1.6 Solutions to Defend Against Sinkhole Attacks

An algorithm which detects sinkhole attacks is presented in [118]. Proposed algorithm first finds a list of suspected nodes, and then effectively identifies the intruder in the list through a network flow graph.

C.1.7 Solutions to Defend Against Sybil Attack

To detect the Sybil attack, two methods were discussed in [151]. One method is radio resource testing in which each node assigns a unique channel to each of its neighbors, including fake neighbors,

and tests whether its neighbors can communicate with it through the assigned channels. Because the radio of a sensor platform is usually incapable of simultaneously sending or receiving on more than one channel, the failure of communication through one channel may be a sign of the Sybil attack. The other method is to use the ID-based symmetric keys. For example, each sensor node is preloaded with a set of keys that are selected from a global key pool by its node ID. The ID of a suspected node is challenged by a set of validating nodes based on the keys shared between the suspected node and the validating nodes. Several other methods were suggested in [151], including registration, position verification, and code attestation.

To defend against Sybil attacks, the identities of every node should be verified. This can be done either directly or indirectly. In direct validation a node directly verifies whether the identity of a neighboring node is valid. For example, a node may assign each of its neighbors a separate channel to communicate and ask them to transmit during a period. Then it checks these channels in a random order within that period. If a node is transmitting in its assigned channel, the node is a physical node. If no transmission is detected on a channel, it indicates that the node assigned to that channel may not be a physical node [151].

In indirect validation another trusted node provides the verification for the identity of the node. For example, every node may share a unique key with the base station. When two nodes need to establish a link between them, they verify each other's identity through the base station by using these keys [133]. At the same time they can be assigned a session key. Nodes can also be allowed to establish links with a limited number of neighboring nodes. Thus, compromised nodes can only communicate with a limited number of verified neighboring nodes, which also limits the impact of Sybil attacks.

Moreover, ID-based public keys [144] also can defeat the Sybil attack because both the ID and location information were taken into the generation of key material during the initialization phase, hence multiple identities need multiple keys, and this is impossible for a malicious node to achieve.

C.1.8 Solutions to Defend Against Wormhole Attacks

Wormholes are difficult to detect because an adversary passes the packets to a distant point from the point at which they are received by using a single hop out-of-band channel. This channel cannot

be listened to by the network. Moreover, the real copy of the packet reaches the point that receives the replayed copy later than the replayed copy. Therefore, the replayed copy is fresher than the real copy.

Detection mechanisms against wormhole attacks can be based on temporal and spatial analysis of the packets. To detect the Wormhole attack, Hu et al. proposed to use packet leashes [137], where location or timing information is embedded in packets, to limit the maximum range over which packets can be tunneled. They require that each node either knows its location or has a tightly synchronized clock so that this information can be used to calculate the maximum distance that a relayed packet could travel.

Directional antennas [152] were also used to defend against the Wormhole attack, where some direction information is used to detect the replayed packets. However, these defenses target ad hoc networks and require expensive hardware devices, which may be infeasible for most resource constrained sensor networks.

Wang and Bhargava [153] proposed to use centralized computing to detect the Wormhole attack in sensor networks, in which a controller collects the location information for all nodes to reconstruct the network topology such that any topological distortion can be visualized. However, the visualization approach incurs too much communication overhead, especially when malicious nodes move around in the entire network because each location change of the Wormhole triggers a new round of execution of the topology reconstruction algorithm. Location-based keys [144] also can effectively address the Wormhole attack because each packet is authenticated by the location-based key.

C.1.9 Summary of the Solutions

Table C.2 summarizes the attacks and proposed solutions related to corresponding attacks. Among those sinkhole attacks and wormholes pose significant challenges to secure routing protocol design, and it is unlikely there exists effective countermeasures against these attacks that can be applied after the design of a protocol has completed. It is crucial to design routing protocols in which these attacks are meaningless or ineffective. Geographic routing protocols are one class of protocols that holds promise.

Table C.2 Attacks and proposed solutions to defend (detect or prevent) against those attacks.

Attack type	**Proposed Solutions for Detection**	**Proposed Solutions for Prevention**
Eavesdropping	N/A	Link-layer encoding [2, 133, 145–147]
DoS	Swarm intelligence [143]	Usage of spread-spectrum communication [13]
Selective forwarding	Acknowledgement monitoring [148], Reporting packet drops [130], Failure detection framework [150]	Multi-path routing [149]
Sybil	Radio resource testing and ID-based symmetric keys [151]	Identity verification [151], ID-based public keys [144]
Node Replication	Distributed detection [142]	ID-based public keys [144]
Wormhole	Packet Leashes [137], directional antennas [152]	Location-based keys [144], centralized computing [153]
Sink hole	[118]	N/A
HELLO flooding	N/A	Identity verification protocol [133]

An ultimate limitation of building a multi-hop routing topology around a fixed set of base stations is that those nodes within one or two hops of the base stations are particularly attractive for compromise. After a significant number of these nodes have been compromised, all is lost. This indicates that clustering protocols like LEACH [17] where cluster-heads communicate directly with a base station may ultimately yield the most secure solutions against node compromise and insider attacks.

Appendix D

D.1 Author's Other Contributions

During his Ph.D. study, the author (İsmail Bütün) has contributed to the literature on variety of topics which would not be included in this dissertation. Here is a short list of these contributions:

1. *Telecommunications and Networking* Author has many contributions in the field of telecommunications and networking:
    - Worked on "Cooperative MAC protocols for Wireless Ad-Hoc Networks". The research was published in [154].
    - Worked on "Queuing Theory" and evaluated the "Quick Start" algorithm for low-latency networks. The research was published in [155].
    - Conducted research on the effect of "Mobility Prediction" on the performance of Cognitive Radio Networks. The research was published in [156].
    - Worked on network coding. Published our research named "Comparison of Routing and Network Coding in Undirected Network Group Communications" in [157].
    - Worked on scheduling in Wireless Networks. Published our research named "Adaptive Rate Transmission with Opportunistic Scheduling in Wireless Networks" in [158].
    - Worked on delay considerations in Wireless Networks. Published our research named "Delay Considerations with Two-hop Opportunistic Relays in Wireless Networks" in [159].
2. *Cryptography and Network Security:* Author has many contributions in the field of cryptography and network security:
    - Author finalized his Master Thesis named "A Blind Digital Signature Scheme using Elliptic Curve Cryptography" and this work is accepted for publication in [160].

Appendix D (Continued)

- Author and his colleagues focused on energy-efficient access control schemes for WSNs and published their work in [25].
- Author conducted research in Siemens Corporate Research Center in 2011 for 6 months. The result of this research was a patent on Patch Management Systems named "Networking Elements as a Patch Distribution Platform for Distributed Automation and Control Domains" [44].

ABOUT THE AUTHOR

İsmail Bütün received his B.Sc. and M.Sc. degrees in Electrical and Electronics Engineering from Hacettepe University, Ankara, Turkey, in May 2003 and October 2006, respectively. He received his second M.Sc. degree in Electrical Engineering from University of South Florida, Tampa, FL, in 2009. He is a member of the Interdisciplinary Communications and Signal Processing Group (for more information regarding *i*CONS group, please visit official web site: `http://icons.eng.usf.edu`) at University of South Florida and he is currently pursuing his Ph.D. degree in Electrical Engineering Department at the University of South Florida, Tampa, FL.

Author's research interests include computer networks, wireless communications, cryptography, network security, security in wireless sensor networks and mobile ad-hoc networks. His contributions to the academic literature (that are not included in this dissertation) are listed in Appendix D.

For further details, please visit his personal web page:

`http://www.eng.usf.edu/~ibutun`