ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ. «Ασφάλεια Ψηφιακών Συστημάτων»

ΑΚΑΔΗΜΑΪΚΟ ΕΤΟΣ 2018-2019

(ΨΣ-ΑΦ-888) ΜΕΤΑΠΤΥΧΙΑΚΗ ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

"Datasets for Intrusion Detection for Wireless Body Area Networks "

Supervisor: Professor Dr. Sokratis Katsikas, University of Piraeus
Author: Asimakopoulos Vasileios MTE1605

Athens, February 2019

ACKNOWLEDGEMENTS

A thesis project is never exclusively the work of one person. Support, ideas and related work come from multiple sources in various ways. In particular, I would like to thank Mr. Sokratis Katsikas for accepting my request to accomplish this Thesis Project under his supervision. His leadership, support and attention to detail have set an example I hope to match some day. I am indebted to him for his trusted advice and constant support.

I am also grateful to the great master program students with whom I have been involved during the academic period and especially to Stelios Gasparinatos, Sotiria Balaoura and Sotiris Stabernas, whose dedication highly motivated me during the last year. Their hard work, organizational and programming skills/knowledge contributed to the implementation and experimental evaluation of several ideas/projects and their feedback was always valuable to me.

Moving towards more personal acknowledgments, I owe special gratitude to my friends for the continuous and unconditional support. Especially to S.Georgoutsos and V.Sakkas, former graduates of the same Master Programme, who supported me from the very first moment I took the decision to apply for it. They have always taken interest in my work and believed in me. They are the persons I troubled the most whenever I faced difficulties and their understanding and encouragement was important to me.

I am, of course, particularly indebted to my parents and my younger sister Elena for their monumental, unwavering support and encouragement on all fronts. They have truly always been there for me, and without them none of this would have ever been actually possible.

Asimakopoulos Vasileios
Athens, February 2019

TABLE OF CONTENTS

1. INTRODUCTION

1.1. Motivation

During the last decade there has been a contentious growth in the research dedicated to the field of wireless body area networks (WBANs), covering a wide range of areas, from theoretical issues to technological advances that made the implementation of such networks possible. These networks use hundreds to thousands of low-cost wireless sensor nodes over an area, for the purpose of monitoring certain natural occurrences and capture geographically distinct measurements over a certain period of time. Nodes employed in sensor networks are characterized by limited resources such as storage, computational and communication capabilities. The power of body area networks, however, lies exactly in the fact that their nodes are so small and cheap to build that a large number of them can be used to cover an extended geographical area.

The spreading interconnection of such devices has given birth to a broad class of exciting new applications in several areas of our lives, including environmental monitoring, healthcare applications, home automation, and traffic control. However, like every network, sensor networks are exposed to security threats which, if not properly detected and addressed, can be particularly dangerous. Their wireless and distributed nature along with the serious constraints in node battery power prevent previously known security approaches to be deployed. This has created a vast number of vulnerabilities that attackers can exploit, in order to gain access in the network and the valuable information transferred within. For example, in an outsider attack, where the attacker node is an unauthorized participant of the sensor network, useless packets may be injected in the network in order to exhaust the energy levels of the nodes, or passively eavesdrop on the network's traffic and retrieve secret information. Even worse, in an insider attack, the attacker has compromised a legitimate sensor node and uses the stolen key material, code and data in order to communicate with the rest of the nodes, as if it was an authorized node. With this kind of man-in-the-middle intrusion, an attacker can launch more powerful and hard to detect attacks that can disrupt or paralyze the network.

Securing wireless body area networks (WBANs), via intrusion detection against similar threats, is a challenging research area necessary for commercially attractive deployments. Unfortunately, while sensor networks were at their early stages, the main research focus was on making sensor networks feasible and useful and less emphasis was placed on security or detection needs. Most of the operating system protocols are built assuming a trusted environment and are very vulnerable against security attacks. Addressing all various kinds of vulnerabilities can become very complex. Different applications employ different types of protocols, thus different types of attacks and weaknesses occur, that require different security mechanisms. Therefore, scientific society should focus not only on how to secure sensor networks, but also on how this can be done through generic and independent solutions, in cooperation with effective intrusion detection methods.

1.2. Problem Statement

1.2.1. WBANs for Telemedicine and M-Health

The development of the BASN is derived from the recent development of the wireless body area networks (WBANs), which is a system that interconnects devices or sensors, worn on or implanted in the human body, in order to wirelessly share information and resources between the devices. Although the two terms may appear to be very similar to each other, the description of the second one (WBAN) is definitely preferred when referring to the type of wireless networks in telemedicine and m-health where each node comprises a biosensor or a medical device with a sensing unit broadcasting through a non-wired medium.

The WBAN was initially proposed to connect personal consumer electronic devices for the sake of convenience to the user. However, it is found to be practically essential in telemedicine and m-health because several sensors placed at different body parts are often required in the cases of many patients who need long-term and continuous collection of medical data. This gives at least two reasons for setting up a WBAN. The first is to optimize the use of resources in order to satisfy the strict constraints in the terminals.

For example, medical data collected from different sensors can be centralized before being passed on to external networks for remote analysis, diagnosis, or treatment. Second, a WBAN enhances the controlling, scheduling, and programming of the overall system such that it is adaptive to body condition and external environment. For example, some nodes of a WBAN may have to be reprogrammed from time to time (e.g., a device for drug delivery). In short, the need to develop a WBASN is driven by the increase in the number of wearable or implanted biosensors to be placed on users.

Figure 1 illustrates the general system architecture and service platform:

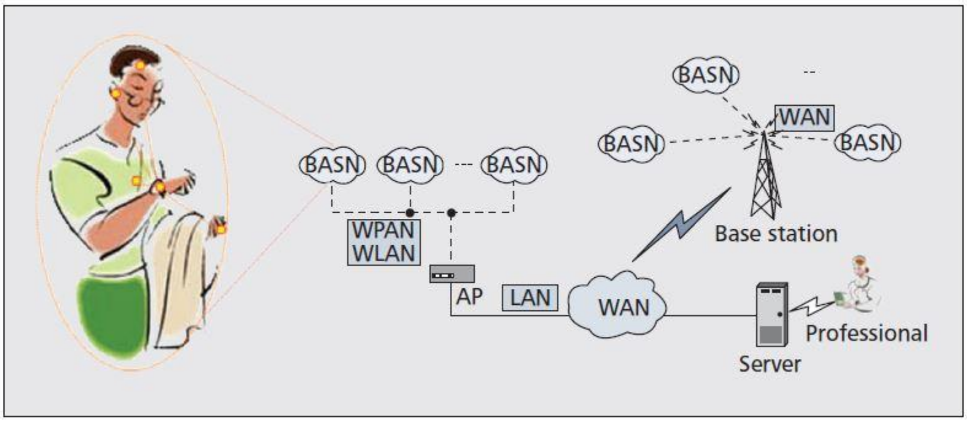


Figure 1 - System architecture and service platform of a WBAN for telemedicine. Image from Carmen C.Y.Poon and Yuan-Ting Zhang: "*A Novel Biometrics Method to Secure Wireless Body Area Sensor Networks for Telemedicine and M-Health*", IEEE Communications Magazine, p.74,§1, April 2006

1.2.2. Intrusion Detection and datasets in WBANs

One of the major security issues in the field of WBAN's, is the broadcast nature of the transmission medium (air). This fact makes collected information even more vulnerable than in wired communications. Thus, security mechanisms such as encryption and authentication are essential to protect information transfers. However, existing network security mechanisms are not yet implemented in this domain, given the limited processing power, storage, bandwidth and energy resources. Public-key algorithms, such as RSA are undesirable, as they are quite expensive. Instead, symmetric encryption / decryption algorithms and hashing functions are significantly faster and constitute the basic tools for securing sensor networks communications. Encryption and authentication mechanisms provide reasonable defense for mote-class outsider attacks. However, cryptography is inefficient when dealing with laptop-class and insider attacks. This fact remains an open problem for additional research and development. The presence of insiders significantly reduces the effectiveness of link layer security mechanisms. This is because an insider is allowed to participate in the network and have complete access to any messages routed through the network and is free to modify, suppress, or eavesdrop on the contents.

There are several classical security methodologies so far that focus on trying to prevent these intrusions. However, it is impossible, or even not practical, to guarantee perfect prevention. Not all types of attacks are known and new ones constantly appear. As a result, attackers can always find security holes to exploit in order to gain access in the sensor network. These intrusions will go unnoticed and they will likely lead to failures in the normal operation of the network as Figure 2a suggests:

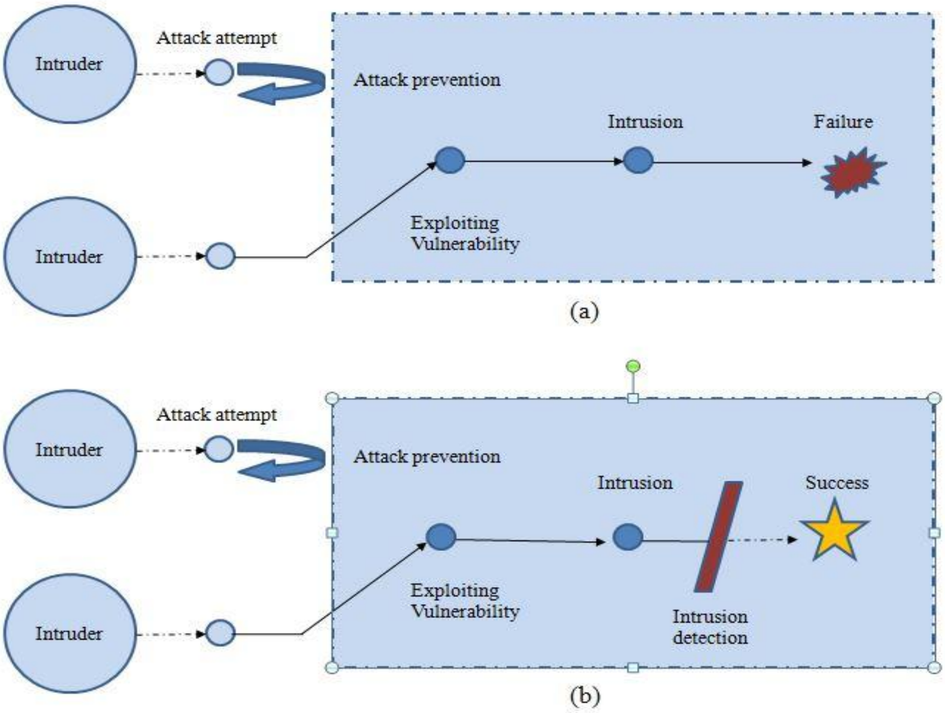


Figure 2 - (a) Attackers may exploit a vulnerability and intrude into the network causing a failure. (b) Intrusion detection counts as a second line of defense.

Because the process of avoiding or preventing security threats cannot be always successful, an Intrusion Detection System (IDS) is needed to detect known and unknown attacks and alert sensor nodes about them. It can act as a second line of defense, by detecting third party break-in attempts, even if this particular kind of attack has not been experienced before. If the intruder is detected soon enough, administrators or users can take appropriate measures before any damage is done or any data is compromised (Figure 2b). An effective intrusion detection system (IDS) can also help scientific society design better prevention mechanisms, by collecting information about new intrusion techniques and attack patterns.

In fact, an IDS allows detecting suspicious or abnormal activities and triggers an alarm when an intrusion occurs. The implementation of IDSs for WBANs is more difficult than other systems, because sensor nodes are usually designed to be tiny and cheap and they do not have enough hardware resources. Additionally, there is no specialized dataset that contains normal profiles and attacks in WBAN that can be used to detect an attacker signature. Considering the above challenges, there are mainly two conditions while designing IDS for WBANs: The IDS must be of high accuracy degree in detecting an intruder that includes unknown attacks, and it also must be lightweight to ensure minimum overhead on the infrastructure of WBANs. Last but not least, an IDS has to be compatible with the characteristics of WBANs and capable of detecting the largest possible number of security threats. The research field of IDS in wireless body area sensor networks is still under development. There are some attempts that concentrate on specific attacks, but not a generalized approach that can be both realistic and lightweight enough to run on computationally and memory restricted devices, such as the nodes of a sensor network.

In this thesis, iDetect, an intelligent IDS Architecture is described, along with two WBAN simulators: 'Castalia' and 'Network Simulator 2' (NS-2). In addition, a specialized dataset for WBAN's is reviewed, to help better detect and classify four types of Denial of Service (DoS) attacks: Blackhole, Grayhole, Flooding, and Scheduling attacks. To achieve this goal, the use of LEACH protocol, one of the most popular hierarchical routing protocols in WBAN's, is presented. A scheme has been defined to collect data from Network Simulator 2 and then processed to produce twenty three features. The collected dataset is called WSN-DS. Artificial Neural Network (ANN) has been applied on the dataset to detect and classify different DoS attacks. The results showed that WSN-DS improved the ability of the IDS to achieve higher classification accuracy rate.

2. SECURITY & BACKROUND IN SENSOR NETWORKS

The design of many wireless body area sensor network applications or lower layer protocols, assume that all nodes are cooperative and trustworthy. This is unlikely to happen in most cases of real world deployments, where nodes are exposed to many threats that can seriously damage the total network functionality. There are many attacks designed to take advantage of the unreliable communication channels and the unattended sensor nodes.

Most sensor networks actively monitor their whereabouts, and it is often easy to work out information other than the monitored data. Such information leakage often results in loss of privacy for the people in the environment. Moreover, the wireless communication employed by sensor networks, supports eavesdropping and packet injection by any possible person-attacker. The combination of these factors demands security for sensor networks to ensure operational security, secrecy of sensitive data and privacy for people in wireless body area network environments.

2.1. Security in Wireless Body Area Networks

According to [7], "because WBAN systems and their supporting infrastructure are geographically distributed, they present a greater challenge in the areas of throughput, data integrity, and data security when compared to traditional clinical systems. Apart from the engineering issues of just ‘making it work,’ there are issues of patient protection that become important. These issues refer to security, which addresses system viability in the areas of safety, security, reliability, fault tolerance, accuracy, repeatability, and human factors".

Patient and data protection require the combination of services to verify the identity of the WBAN wearer/user (i.e., authentication), protect the communication confidentiality, establish secure tunnels between the wearer and their personal devices, maintain the integrity of sensor data from its birth to final storage and finally deter access to stored data or data in transmission.

2.1.1. Security Requirements

More specifically, the variety of security threats in WBANs, demands specific security requirements in these networks, for efficient performance. These requirements are described according to [2]:

- "*Confidentiality*: Owing to the open nature of wireless medium, anyone is capable of eavesdropping on the insecure wireless channel and obtaining unauthorized access to confidential patient information. In WBANs, it is vital to protect the sensed information from the body sensors as well as the communication transmission between different wireless sensor nodes in the network. Hence, data confidentiality is a primary security goal in wireless based healthcare systems.

- *Authentication*: Since patients do not wish that their medical information be accessible by anyone other than their doctors and medical professionals, it is crucial that only authorized entity in the network gets access to retrieve and review medical records. An unauthenticated entity in the network may cause serious damage by falsifying medical and/or patient data. It is necessary that every node in the network is authenticated to prevent malicious intruders from gaining access to sensitive data.

- *Integrity*: It is also essential in the medical field that data in the network does not get modified or altered in any way. If a malicious node in the network is successful in changing and falsifying data, it can result in disastrous consequences. For instance, altered medical data may result in wrongful diagnosis or treatment of patient. Data integrity is thus a security requirement paramount to the performance of the healthcare system.

- *Access control*: The healthcare organization is a large system with several networks and groups of people working in different sectors. Even if an entity in this network is authentic it is necessary to clearly define and restrict data access according to their roles and responsibilities. For instance, a medical doctor and hospital staff should not have same access to the same patient data. In case of a lack of proper access and permissions, a compromised employee in the system can steal critical financial and/or medical information.

- *Privacy*: Healthcare systems store huge volumes of data about patients, doctors, staff, insurers and many others. Most of the data stored in these networks include personal information about people, which even if not financially valuable violates the privacy of an individual".

2.2. Threat Models

The goal of an attacker, either he is insider or outsider, is to directly manipulate user data, or try to gain access to routing topology. What makes it even easier for him is the fact that most protocols for wireless sensor networks are not designed to bear security threats in mind. As a consequence, deployments of sensor networks rarely include security protection mechanisms and little, or zero effort is usually required from the side of the attacker to perform the attack.

In sensor networks security, an attacker can perform a wide variety of attacks. Not all of the attacking parties have the same goals or motivations. Therefore, in order to organize and design better defense systems, there is a threat model that discriminates between two types of attacks: insider and outsider attacks.

2.2.1. Insider Attacks

From a security point of view, an insider attack is considered by many more dangerous. The malicious person physically captures a node and reads its memory. This way, he can obtain its key material and counterfeit node messages.

Having access to legitimate keys, the attacker can launch several kinds of attacks without easily being detected:

- *False data injection (stealthy attack)*: the attacker injects false aggregation results, which are significantly different from the true results

- *Selective reporting*: the attacker stalls the reports of events that do happen, by dropping legitimate packets that pass through the compromised and controlled node.

Of course, an adversary should not have unlimited capabilities. There is some cost associated with capturing, reverse-engineering and controlling a node. This fact affects the design of security protocols, as it is easier to offer some protection against a few compromised nodes, but not for the case where a large portion of the network is in control of the attacker.

2.2.2. Outsider Attacks

In an outsider attack (in most cases node intruder attack), the attacker node is not an authorized member of the wireless sensor network. Authentication and encryption techniques prevent such an attacker to gain any special access to the sensor network. The intruder node can only be used to launch passive attacks, like the following:

- *Passive eavesdropping*: The attacker eavesdrops and records (saves) encrypted messages. The messages may then be analyzed in order to discover secret keys.

- *Denial of service attacks*: In its simplest form, an adversary attempts to disrupt the network's operation by broadcasting high-energy signals. In this way, communication between legitimate nodes could be jammed, or even worse, totally energy consumed.

- *Replay attacks*: The attacker captures messages exchanged between legitimate nodes and replays them in order to change the amassing results.

There are more sophisticated attacks that exploit specific characteristics of the routing protocols in order to effect the topology and gain access to the routed information. Some of them are described below:

2.2.3. The Sinkhole Attack

The sinkhole attack is a particularly severe attack that prevents the base station from obtaining complete and correct sensing data, thus forming a serious threat to higher layer applications. In a sinkhole attack, a compromised node tries to draw all or as much traffic as possible from a particular area, by making itself look attractive to the surrounding nodes with respect to the routing metric. As a result, the attacker manages to attract all traffic that is destined to the base station. By taking part in the routing process, he can then launch more severe attacks, like selective forwarding, modifying or even dropping the packets coming through.

For instance, several vulnerabilities of two popular routing protocols of sensor networks, namely the MintRoute and the MultiHopLQI were discovered from the scientific society. The vulnerabilities showed how these protocols can be exploited by an attacker to launch a sinkhole attack. It is very easy for the adversary to make the compromised node look attractive to its neighbors, or make them look less attractive and eventually make all nodes choose that specific node as their new parent.

2.2.4. The Wormhole Attack

The wormhole attack is a severe threat against packet routing in sensor networks that is particularly challenging to detect and prevent. To launch such an attack, an adversary establishes a low-latency link, referred as a wormhole link, between two points of the network, as shown in Figure 3. Once the wormhole link is operational, the adversary eavesdrops messages at one end and tunnels them (possibly selectively) to the other end, where the packets are retransmitted. The low-latency link used in this attack, as well as any devices attached at each end of the link, belong only to the attacker and are not compromised resources of the network. The link is realized in such a way that packets can travel from one end to the other faster than they would normally do via a multi-hop route in the network. The sensor nodes cannot detect the existence of such a link, as it can be realized with other means, such as a wired connection or an out-of-band wireless transmission.

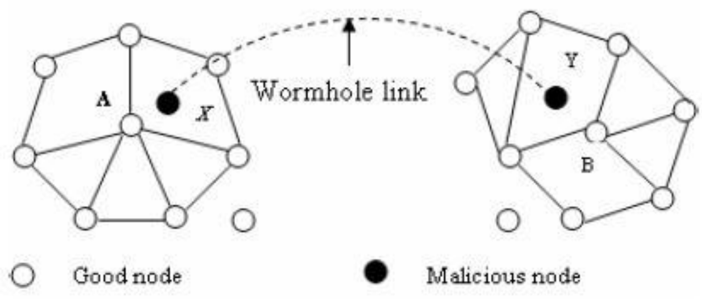


Figure 3 - A wormhole attack between two separate points of the network. Image from Khin Sandar Win: *"Analysis of Detecting Wormhole Attack in Wireless Networks"*, World Academy of Science, Engineering and Technology International Journal of Electronics and Communication Engineering Vol2, No12, p,§II.A,2008

As shown in the example of Figure 3, the pure effect of the wormhole attack is that the nodes within region A think they are neighbors with the nodes within region B and vice versa. If the attacker carefully chooses the place of the wormhole's end-points, then it can use it to completely disrupt routing and attract a significant amount of traffic. Therefore, if one end of the wormhole is close to the base station, then nodes situated multiple hops away could be convinced that they are only one or two hops away. As a result, these nodes will choose to use the high quality link for their transmission enabling other kind of attacks such as the sinkhole attack.

2.2.5. The Sybil Attack

A Sybil attack is one in which an attacker uses a malicious device to create a large number of pseudonymous entities, using them to gain disproportionately big influence. Sybil nodes are referred as a malicious device's additional identities. Newsome et al. in [8], introduce a "taxonomy of the different forms of the Sybil attack in sensor networks. In terms of communication, Sybil nodes can communicate directly or indirectly with legitimate nodes. In the second case, legitimate nodes are able to communicate with the Sybil nodes through the malicious device, which claims to be able to reach the Sybil nodes. Moreover, the mentioned device can fabricate a new identity for a Sybil node, or it can steal an identity from a legitimate node. Finally, in terms of time, the attacker may try to have the Sybil identities participate in the network all at once, or present a large number of identities over a period of time."

Sybil attack can be used against many protocols in sensor networks. In multipath routing, if a geographic routing protocol is used, a Sybil node could appear in more than one place at once, instead of having one set of coordinates. In-network processing is also vulnerable to Sybil attack. An attacker can also affect aggregation results of sensor readings, or a voting process amongst sensor nodes and make the system come to wrong conclusions. Therefore, Sybil attacks can become a significant threat to the normal operation of a wireless sensor network.

2.2.6. The HELLO Flood Attack

Many WBAN protocols require nodes to broadcast HELLO packets for neighbor discovery purposes. After just a few messages have been exchanged, most nodes have a complete picture of their immediate vicinity and a routing topology logically forms in a self-organizing way. However, if a laptop-class attacker broadcasts such packets with large enough transmission power, he could convince every node in the network that the adversary is its neighbor and advertise attractive routing pathways through itself. After convincing portions of the network that it is truly the best routing option, it might choose to ignore incoming messages, effectively disabling large portions or even the entire network.

Unlike the rest of attacks we described so far, the HELLO flood attack does not require an attacking node to create legitimate traffic to be successful. So, for example, even an outsider attacker can capture legitimate "HELLO" messages as they breezed through the air and then forward them with a more powerful antenna. Those messages would reach other nodes well beyond the actual reach of a real sensor node's hardware. this kind of forwarding and redistribution leads to false network topologies and fake routing information.

2.3. Realistic Obstacles to Sensor Network Security

Although wireless body area sensor networks have an ad-hoc nature, there are several limitations that make security mechanisms proposed for ad-hoc networks not applicable in this setting. In particular, security in sensor networks is complicated by more constrained resources and the need for large-scale deployments. A summary of these limitations follows below:

2.3.1. Constrained Hardware

A wide range of sensor node platforms has emerged over the past years. So far, for such devices, the trend has been to increase the lifetime of the nodes by decreasing the resources such as memory, CPU, and radio bandwidth. Establishing secure communication between sensor nodes becomes a challenging task, given these limited resources, as well as the lack of control of the wireless communication medium. Public-key algorithms, such as RSA or Diffie-Hellman key agreement are undesirable, as they are computationally expensive. Instead, symmetric encryption/decryption algorithms and hash functions are even more preferable.

2.3.2. Wireless Communication

Sensor nodes communicate through wireless communication, which is particularly expensive from an energy point of view (one bit transmitted is equivalent to about a thousand CPU operations). Therefore, one cannot use complicated protocols that involve the exchange of a large number of messages. Additionally, the nature of communication makes it particularly easy to eavesdrop, inject malicious messages into the wireless network or even hinder communications entirely using radio jamming.

2.3.3. Exposure to Physical Attacks

Unlike traditional networks, sensor nodes are often deployed in areas easily accessible by an attacker, presenting the risk of physical attacks that can expose their cryptographic material or modify their hidden code. This problem is magnified further by the fact that sensor nodes cannot be made tamper-proof due to increases in hardware cost. Therefore, sensor nodes are more likely to suffer a physical attack in such an environment, compared to typical PC workstations which are located in a secure place and mainly face attacks from a network.

2.3.4. Large Scale Deployment

Future sensor networks will have hundreds to thousands of nodes so it is clear that scalability is a prerequisite for any attempt in securing sensor networks. Security algorithms or protocols that were not designed with scalability in mind offer little or no practical value to sensor network security.

2.3.5. Aggregation Processing

An effective technique to extend sensor network lifetime is to limit the amount of data sent back to reporting nodes since this reduces communication overhead. However, this cannot be done unless intermediate sensor nodes have access to the exchanged data to perform data fusion processing. End-to-end confidentiality should therefore be avoided as it hinders aggregation by intermediate nodes and complicates the design of energy-aware protocols.

3. THE INTRUSION DETECTION PROBLEM

3.1. Introduction

The existing techniques can effectively protect wireless sensor networks against certain attacks. However, the security protocols they use, are based on a particular attacker assumption. If the attacker is "weaker", any security protocol will achieve its security goal; it will prevent an intruder from breaking into a sensor network and alter its proper operation. If, however, the attacker has "stronger" capabilities there is a high probability that the adversary will break in. Because of their resource constraints, sensor nodes usually cannot deal with very strong adversaries. So what is needed is a second line of defense: An Intrusion Detection System (IDS) that can detect a third party's attempts of exploiting the vulnerabilities of the network, even if such attacks have not been experienced before.

Intrusion detection systems provide a necessary, in-depth protection layer zone for wired networks. However, wireless detection is not crystal clear in the context of such networks. In this chapter a definition of the problem is given and several scenarios are presented , assuming one attacking node. It is essential to set the theoretical foundation of this research area first, before trying to design and implement an Intrusion Detection System (IDS) specifically for sensor networks.

Furthermore, the existing intrusion detection techniques from wired networks are surveyed and important approaches appropriate for wireless sensor networks are then indicated. In Section 3.4, we briefly survey some of these approaches, i.e. one of which is the watchdog approach. After that, the requirements an IDS for sensor networks are outlined and the existing bibliography is reviewed. Last but not least, necessary and sufficient conditions under which an IDS can successfully detect an attacker are described, along with scenarios in which cooperative intrusion detection is unsolvable. In the paragraph to follow, there will be a review of designing an IDS for Sensor Networks.

3.2. Designing an IDS for WBANs

In intrusion detection, the main goal is to provide an automated mechanism that identifies the source of an attack and generates an alarm to notify the network or the administrator, so that appropriate preventive actions can take place. As an attack it is considered any set of actions that target the computing or networking resources of our system. Attackers may be using an external system without authorization or have legitimate access to our system but are abusing their privileges (i.e., an insider attack). It is important to realize here that the IDS comes into surface after an intrusion attempt has occurred. It does not try to prevent these attempts in the first place.

IDS has become an important security component of WBANs; however, the implementation of IDS in WBANs introduces numerous challenges that can have negative impact on the whole system and network performance. It is inefficient to use IDS in every sensor node due to their resource-constrained nature. IDS components should be installed in places where sensor nodes can be able to defend against certain threats of the network. IDS is also used in WBANs where extremely large amount of traffic is transmitted. This is because wireless sensor nodes, generally have restrictions in handling huge data in the network and there is a possibility an intrusion to be missed.

According to [28], "there are two main components of IDS, features extraction and modeling algorithm. Features extraction defines measured attributes that are linked to the IDS functionalities. Modeling algorithm is the main component; the accuracy and the efficiency of detecting and responding to intrusions depend on the modeling algorithm. IDS may have components that depend on the network characteristics and possible intrusions. Most of IDSs have six common components as shown in Figure 4":

- *Monitoring component*: This is used for local activity monitoring or for monitoring neighbor sensor nodes. This component mostly monitors internal activities, traffic patterns, and resource utilization.

- *Analysis component*: It contains all records of normal and abnormal behaviors for all nodes in the network as described in [35].

- *Detection component*: This is the main component that is built upon the modeling algorithm. It works after analyzing network behaviors. Decisions are made to determine whether such behaviors are malicious or not, according to [36].

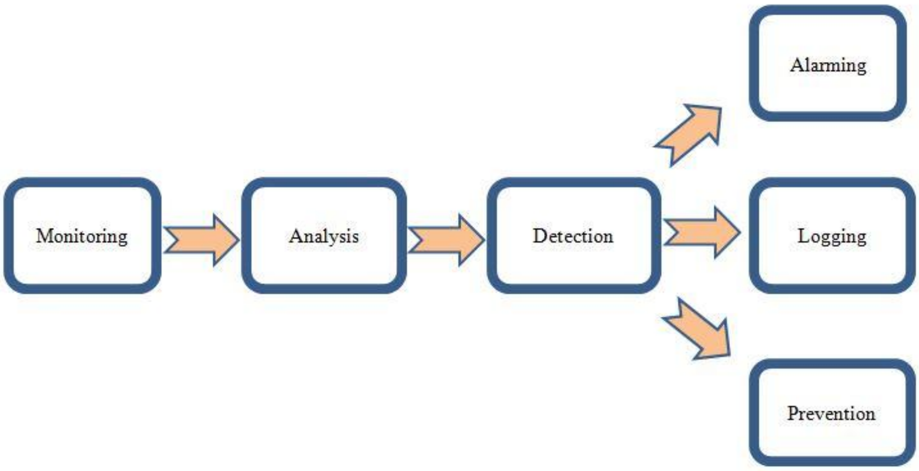


Figure 4 - IDS components as described in *"WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks"*[28], p4 §2.3.

As described in [37], "the other three components of IDS consist of actions that can be taken, either one, two, or all of them:

- *Logging*: storing each packet in a log file so that security administrator can use it for later analysis.

- *Alarming*: a responding generating component in case of detection of an intrusion. The response may trigger an alarm to announce the misbehaving node(s).

- *Prevention*: an advanced step that can be added to IDS to enable it to take an action to prevent dealing with an attack once detected. This can be done, for example, by excluding harmful nodes from the network**.**"

3.2.1. Intrusion Detection Techniques

In order to detect an intruder, it is necessary to use an equivalent intrusion detection model. It is also essential to know what an IDS should look out for. In particular, an IDS must be able to distinguish the difference between normal and abnormal activities (or else norms) in order to discover malicious attempts in time. However this can be difficult, since many behavior patterns can be unpredictable and unclear. There are three main techniques that an intrusion detection system can use to classify actions.

- *Misuse detection*: In misuse detection or signature-based detection systems the observed behavior is compared with known attack patterns (signatures). So, action patterns that may cause a security threat must be clearly defined and given as input to the system. The misuse detection system tries to recognize any "suspicious" behavior according to these patterns. Any action that is not clearly prohibited is allowed. The main disadvantage of such systems is that they cannot detect newly created attacks. Someone must constantly update the attack signature database. Another difficulty is that signatures must be written in a way to include all possible combinations of the relevant attack, and yet avoid flagging non-intrusive activity as an intrusive one (false positive/false negative schemes).

- *Anomaly detection:* Anomaly detection copes together the limitations of misuse detection by focusing on normal behaviors, rather than attack behaviors. This technique first describes what consists of a "normal" behavior (usually established by automated training) and then flags as intrusion attempts any activities opposing from this behavior by a statistically significant amount. In this way there is a considerable possibility to detect novel attacks as intrusions. There are two problems connected with this approach: First, a system can exhibit legitimate but previously unseen behavior. This would lead to a substantial false alarm rate, where anomalous activities that are not intrusive are flagged as intrusive (false positive scheme). Secondly and even worse, an intrusion that does not exhibit anomalous behavior may not be detected, resulting in false negatives, an obviously non-wanted scenario.

- *Specification-based detection*: This kind tries to combine the strengths of misuse and anomaly detection. It is based on deviations from normal behavior. However, in this case, the normal behavior is not defined by machine learning techniques and training. It is based on manually defined specifications that describe what is a correct operation and monitors any behavior with respect to these constraints. In this way, legitimate but previously unseen behaviors will not cause a high false alarm rate, as in the anomaly detection approach. Also, since it is based on deviations from legitimate behaviors, it can still detect previously unknown attacks. On the other side, the development of detailed specifications by humans can be time-consuming and bare the inherent risk that certain attacks may pass undetected.

Caution must be taken when applying the anomaly detection techniques in wireless sensor networks. It is not easy to define what is a "normal behavior" in such systems, as they usually adapt to variations in their environment or according to other parameters, such as the remaining battery level. So, these legitimate changes of behavior may easily be mistaken from the IDS as intrusion attempts. Moreover, sensor networks cannot bear the overhead of automatic training, due to their low energy resources. Specification-based detection, seems the most appropriate approach in this case, if one can design appropriate rules that cover as broad range of attacks as possible.

3.2.2. Intrusion Detection Architectures

Traditionally, intrusion detection systems for fixed networks were divided into two categories: host-based and network-based. The host-based architecture was the first architecture to be explored in intrusion detection. A host-based intrusion detection system (HIDS) is designed to monitor, detect and respond to system activity and attacks on a specific host (node). Any decision made is based on information collected at that host by reviewing audit logs (i.e. raw files) for suspicious activity. This conflicts with the distributed nature of sensor networks and makes it impossible to detect network attacks. A network-based architecture is more appropriate in our case.

Network-based intrusion detection systems (NIDS) use raw network packets as the data source. A network-based IDS typically listens on the network, and captures and examines individual packets in real time. It can analyze the entire packet, not just the header. In wired networks, active scanning of packets from a network-based intrusion detection system is usually done at specific traffic concentration points, such as switches, routers or gateways. On the other hand, wireless sensor networks do not have such "bottlenecks". Any node can act as a router and traffic is usually distributed for load balancing purposes. So, it is impossible to monitor the traffic at certain points.

So, when designing an IDS for sensor networks, we must be careful of where to locate the detection agents, due to the distributed nature of the network and traffic routed within. One possible solution is to have an identical agent inside every node. That would be a realistic solution, if the agents were designed to be lightweight and cooperative through a distributed algorithm. Another solution would be to have a hierarchical model, where some more computationally intensive agents were placed on certain nodes, while other agents with restrictive tasks were placed on the rest of the nodes.

3.2.3. Decision Making Techniques

Intrusion detection systems can be further classified according to the decision making techniques that they use in order to detect and initiate a response to an intrusion attempt. This decision can be made either collaboratively or independently by the nodes.

According Zhang et al. [9],"since the nature of sensor networks is distributed and most of the services provided require cooperation of other nodes, it is only natural that intrusion detection should also be done in a cooperative manner. In this case, every node participates in intrusion detection and response by having an IDS client installed on them. Each node is responsible for detecting attempts of intrusion locally. If an anomaly is detected by a node with weak evidence, or if the evidence is inconclusive, then a cooperative mechanism is initiated with the neighboring nodes in order to take a global intrusion detection action." More sophisticated cooperative decision-making schemes, may use mobile agents or fuzzy logic to better support the decision process.

When designing a cooperative decision making mechanism for intrusion detection in sensor networks, one should consider the fact that a node can be compromised and hence, send false data to its neighbors trying to affect the decision. So, one must be skeptical as to which nodes to trust. The fact that it is difficult for an adversary to compromise the majority of the nodes in a specific neighborhood can play an important role here. Moreover, a cooperative mechanism has to consider the bandwidth and energy resources of the nodes. The nodes cannot exchange security data and intrusion alerts without considering the energy that has to be spent for sending, receiving and processing these messages.

In an independent decision-making system, there are certain nodes that have the task to perform the decision-making functionality. They collect intrusion and anomalous activity evidences from other nodes and based on them they can make decision about network-level intrusions.

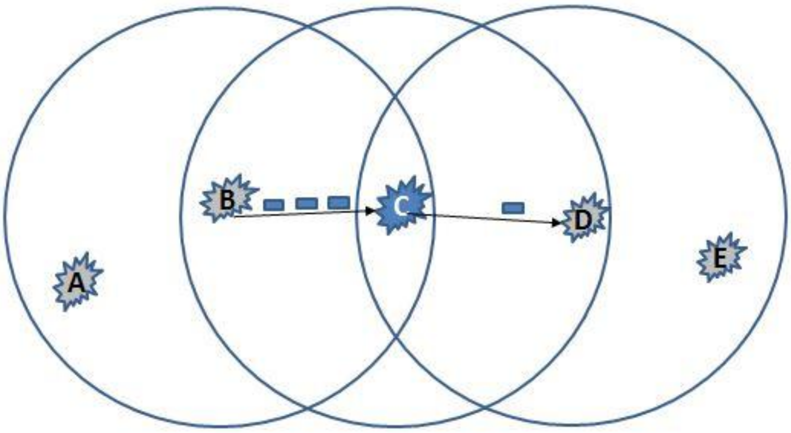


Figure 5 - Note B is selectively forwarding packets to node C. Node A eavesdrops to node B's transmissions.

The rest of the nodes do not participate in this decision. In such architectures, the decision-making nodes can attract the interest of an attacker, since their elimination would leave the network undefended. Furthermore, the information that they process is limited, since it originates from specific nodes. Another disadvantage of such approaches is that they restrict computation-intensive analysis of overall network security state to a few key nodes. Their special mission of processing the information from other nodes and deciding on intrusion attempts results in an extra processing overhead, which may quickly lead to their energy exhaustion, unless different nodes are dynamically elected periodically.

3.3. Requirements of IDS for WBANs

In order to explain more on the requirements that an IDS system for WBANs should satisfy, one has to look at the specific characteristics of these networks. Each sensor node has limited communication and computational resources and a short radio range. Furthermore, each node is a weak unit that can be easily compromised by an adversary, who can then load malicious software to launch an insider attack.

In this context, taking under consideration a distributed architecture based on node cooperation, it could be a desirable solution. In particular, an IDS for wireless sensor networks must satisfy the following properties:

- *Localize auditing*. An IDS for sensor networks must work with localized and partial audit data. In such networks there are no centralized points (apart from the base station) that can collect audit data for the entire network, so this approach fits the sensor networks example. Dealing with partial data means that the IDS should also address the problem of high false alarm rate.

- *Minimize resources*. An IDS for sensor networks should utilize a small amount of resources. The wireless network does not have stable connections and physical resources of network and devices such as bandwidth and power, are limited. Disconnection can happen at any time. In addition, the communication between nodes for intrusion detection purposes should not take too much of the available bandwidth.

- *Trust no single node*. In a collaborative IDS, the nodes cannot assume that other participant nodes can be trusted. Unlike wired networks, sensor nodes can be easily compromised. These nodes may behave normally with respect to the routing of the information in order to avoid being detected by the IDS. However, they can expose a malicious behavior to obstruct the successful detection of another intruder node. Therefore, in cooperative algorithms, the IDS must assume that *no* single node can be fully trusted.

- *Be truly distributed*. The process of data collection and analysis should be performed on a number of locations, in order to distribute the load of the intrusion detection. The distributed approach also applies to execution of the detection algorithm and alert correlation.

- *Support addition of new nodes*. In practice it is likely that a sensor network will be populated with more nodes after its deployment. An IDS should be able to support this operation and distinguish it from an attack (e.g. wormhole attack) that has the same effect.

- *Be secure*. An IDS should be able to withstand a hostile attack against itself. Compromising a monitoring node and controlling the behavior of the embedded IDS agent should not enable an adversary to revoke a legitimate node from the network, or keep another intruder node undetected.

3.4. Existing Approaches

As we saw previously, in order to apply a network-based intrusion detection system in wireless sensor networks, packet monitoring should take place in several nodes of the network, due to its distributed nature. In this section we look at a technique that can be used for packet monitoring, called the Watchdog Approach.

According to [13], "the watchdog approach relies on the broadcast nature of the wireless communications and the fact that sensors are usually densely deployed. Each packet transmitted in the network is not only received by the sender and the receiver, but also from a set of neighboring nodes within the sender's radio range. Normally these nodes would discard the packet, since they are not the intended receivers, but for intrusion detection this can be used as a valuable audit source. Hence, a node can activate its IDS agent and monitor the packets sent by its neighbors, by overhearing them. However, this is not always adequate to draw safe conclusions on the behavior of the monitored node. Furthermore, to detect certain attacks, it is not enough to monitor just one node, but rather a link, meaning the packets transmitted by the nodes at both of its ends. For example, to detect selective forwarding, a watchdog should be able to overhear packets arriving at a node and transmitted by that node. "

Someone could argue that the watchdog approach increases the energy consumption of the nodes, since they have to overhear packets not destined for them. However, in most radio stacks of today's sensor platforms each node receives packets sent by neighboring nodes anyway. They cannot know if a packet is addressed to them unless they receive it and check the destination field. So, the only overhead imposed to the nodes is any further processing of the packet.

Several other proposed architectures of intrusion detection systems already exist. The first scheme to be proposed was introduced by Zhang et al.[9], which is a distributed and cooperative IDS model, where every node in the network participates in the detection process. Another architecture, called LIDS[10] utilizes mobile agents on each of the nodes. These agents are used to collect and process data on remote hosts and transfer the results back to their home nodes, or migrate to another node for further investigation. These IDS architectures cannot be applied directly to wireless sensor networks. The differences in the nature of the two kinds of networks impose different requirements, which forces scientific society to design new solutions. A first attempt to apply anomaly detection in wireless sensor networks is presented by da Silva et al.[15] According to the author's proposed algorithm, "there are some monitor nodes in the network, which are responsible for monitoring their neighbors looking for intruders. These nodes listen to messages in their radio range and store certain message fields that might be useful to the rule application phase. The rules concern simple observations, such as:

- the message sending rate must be within some limits,
- the payload of a forwarded message should not be altered,
- the retransmission of a message must occur before a defined timeout, and
- the same message can only be retransmitted a limited number of times."

Then they try to detect some attacks, like message delay, repetition, data alteration, black-hole and selective forwarding. It is concluded from the paper that the buffer size to store the monitored messages, is an important factor that greatly affects the false positives number. Given the restricted memory available in motes, it turns out that the detection effectiveness is kept to lower levels.

Loo et al. in [17] and Bhuse and Gupta [18] describe two more IDSs, emphasizing on routing attacks in wireless sensor networks. Both papers assume that routing protocols can also be applied to WBANs: Loo et al. assume the AODV (Ad hoc On-Demand Distance Vector) protocol, while Bhuse and Gupta use the DSDV and DSR protocols. However, to the best of our knowledge, these routing protocols are not attractive for wireless sensor networks and they have not been applied to any implementation that we are aware of.

3.5. The Intrusion Detection Problem & Conditions for Solving

Intrusion detection not only means to detect that some node has been attacked, it also includes identifying the source of an attack. In this case, the cooperative intrusion detection process is triggered by an attack and the subsequent alerts by the local alert modules of the neighboring sensors. The process ends by having the participating sensors jointly expose the source. The idea of cooperative intrusion detection is to exchange the outputs of local alert modules, thereby narrowing down the set of possible nodes that could be the attacker.

The effectiveness of an IDS depends on determining the most relevant features of network traffic that needs to be constantly monitored, to ensure accurate detection of attacks. Redundant and non-relevant features often increase the overhead and computational complexity of the system. Feature selection is hence one of the most important problems in intrusion detection. Genetic algorithms are known to "provide a simple framework for selecting optimal feature sets to increase detection rate", according to [26].While this solution has been traditionally implemented in wired networks, their adaptability to WBANs is unfortunately not straightforward. It is of paramount importance to identify features that are capable of detecting specific attacks in WBAN without significantly increasing false alarm rates and energy consumption.

3.6. iDetect: Intelligent Intrusion Detection System Architecture

A clever IDS example is iDetect. More specifically, it analyzes the effectiveness of evolutionary learning algorithms (in example genetic algorithms), as applied to IDSs in WBAN's. In general, most of these learning algorithms often have high computational complexity that renders them impractical for use in such resource-constrained networks. The proposed approach in [2], enabled the wireless sensor and mobile nodes that are deployed within the network to use such evolutionary learning techniques for autonomous decision making.

The proposed algorithm in [2] was tested on three different types of attacks targeted at a WBAN:

- "*Jamming attacks*: Jamming attacks seek to prevent either the transmission or reception of data packets on a wireless network. Because they are often easy to implement and costly to defeat, detecting them is key to safe operation of a WBAN. Since constant jamming is often ineffective from an attacker’s point of view, Random Jamming and Deceptive Jamming attacks are better used to run experiments.

- In *Random Jamming* attack, the malicious node in WBAN introduces random noisy transmissions onto the channel, thereby disrupting the communication between wireless body sensors and the gateway device. As a variation of this attack, in Deceptive Jamming, the malicious node constantly injects regular packets onto the wireless channel. This attack deceivingly leads other nodes in WBAN to believe that a normal communication is occurring and are led to remain silent for the duration of communication, while the attacker seizes control of the channel.

- *Selective forwarding attack*: An attack used to delay or prevent the propagation of messages through the network. While relaying critical

> medical information in WBAN, timely transfer of sensor information is key to effective treatment. In targeted attacks such as selective forwarding, the attacker is modeled to forward some information while entirely preventing the transmission of other information."

The effectiveness of the proposed approach was demonstrated by measuring the total number of detections, false positives and false negatives in the system. Results showed that low false positives rate and false negatives rate were achieved.

4. RELATED WORK (SIMULATION - DATASETS)

4.1. Introduction

As reviewed in previous chapters, technological progress in WBAN systems and medical devices equipped with wireless communication interfaces, are revolutionizing the way of inquiring and distributing healthcare. "The integration of body sensors, mobile devices and wireless networking holds great potential for significantly improving the quality of healthcare today. Wireless sensor devices used to monitor vital signs are increasingly becoming part of healthcare applications", according to [21]. In this rapidly rising sector of IoT, medical devices (implanted or wearable's) are used to effectively monitor patients' vital signs such as blood pressure, heart rate, glucose levels, and perform cardiac pacing and deep brain simulation. Health status monitoring, processing and passing information using lightweight and inexpensive wireless sensors, helps provide immediate feedback to patients and medical professionals enabling an effective diagnosis and treatment.

Owing to the sensitive and critical nature of patient health information stored and transmitted in WBANs, any malfunction, vulnerabilities or security threats in these networks is of utmost concern. For instance, security attack on the wireless communication channel or malware on the medical device can result in incorrect data leading to false diagnosis and treatment. In the past decade, targeted attacks on healthcare have grown in massive proportions. Security attacks in WBANs could be targeted either by compromising the body sensor node/device in the network or by attacking wireless network channels. Most of these attacks on wireless channel "are not exclusive to WBAN but also occur in other wireless domains such as ad-hoc and sensor networks", according to [2].

4.2. Existing Security Solutions in BANs

In the past decade, researchers have achieved several ways of enhancing these networks, mainly using cryptographic mechanisms based on symmetric key and public key (asymmetric) cryptography.

CodeBlue project, one of the pioneers in medical monitoring, was intended for pre-hospital and in-hospital emergency care" according to [22]. It was one of the first to "recognize the security and privacy issues in these networks and used TinySec and elliptic curve cryptography (ECC) based on public key encryption scheme for authentication". That system however did not consider any other security issues. ALARM-NET system was developed for dynamic support living and active monitoring. According to [23], it "provides link layer security using AES-based encryption implemented at the hardware". Sensor network for assessment of patients (SNAP) was next designed to address basic security sectors such as privacy, security and integrity in medical sensor networks. This architecture used "multi-level authentication and multiple encryption mechanism using ECC", as seen in [24]. Identity Based Encryption scheme (IBE-Lite) is also used to provide encryption, whereas the server shares the encryption key with all the sensors in the network during initialization stage.

Several other biometric methods have been used for key establishment to secure communications in WBANs. For example, physical biometric values or characteristics are included in popular techniques for encryption, authentication and key generation/distribution in a WBAN. According to [2], "physiological values such as ECG have been evaluated to generate cryptographic key that can be used to establish an authenticated communication channel. Received signal strength (RSS) variation is used to distinguish between signals from sensor devices on the same body from an external signal". Shield, is an external base station that protects implanted wearable devices by instantly receiving and jamming message packets from the devices in such a way, that only the base station is able to decipher the messages. IMDGuard is another biometric-based cryptographic solution used for body devices, where "the patient ECG signals are used for key extraction", as described in [25]. Above methods, however, cannot defend the medical devices against denial of service attacks.

When legitimate nodes in the network are compromised, a second line of defense is required and is often provided by an IDS. While intrusion prevention security measures are widespread in the security architectures proposed for regular BAN's, IDS's in wireless BAN networks are still in their premature stages of development. MedMon for instance, is "a wearable external monotoring device added to the BAN to detect anomalies in the wireless channel", as per [31].

4.2.1. Castalia: A simulator for Wireless Sensor Networks - BANs

Castalia is a simulator for Wireless Body Area Networks (WBANs) and in general networks of low-power implanted devices. Its backbone is the OMNeT++ platform and it can be used by any researcher who wants to test his distributed algorithms in realistic wireless channel and radio models. It achieves a realistic node behaviour especially relating to access of the radio. Castalia can also be used to score different platform characteristics for specific applications, since it is "highly parametric, and can simulate a wide range of platforms". The main features of Castalia according to Athanassios Boulis,*User's Manual*, NICTA, May 2013 are:

- "Advanced channel model based on empirically measured data:
  - Model defines a map of path loss, not simply connections between nodes
  - Complex model for temporal variation of path loss
  - Fully supports mobility of the nodes
  - Interference is handled as received signal strength, not as separate feature
- Advanced radio model based on real radios for low-power communication.

- Probability of reception based on SINR, packet size, modulation type. PSK FSK supported, custom modulation allowed by defining SNR-BER curve.
- Multiple TX power levels with individual node variations allowed
- States with different power consumption and delays switching between them
- Realistic modelling of RSSI and carrier sensing

- Extended sensing modelling provisions
  - Highly flexible physical process model.
  - Sensing device noise, bias, and power consumption.

- Node clock drift
- MAC and routing protocols available.
- Designed for adaptation and expansion."

Castalia was designed in a proper way from its birth, so that the users can easily mount their algorithms and protocols into it, whilst taking advantage of the provided features. "Proper modularization and a configurable, automated build procedure help towards this end. The modularity, reliability, and speed of Castalia is partly enabled by OMNeT++, an excellent framework to build event-driven simulators" according to [38].

On the other hand, it is wrong to consider that Castalia is sensor-platform specific. Its main purpose is to provide a quite plain, reliable and realistic framework for the first evaluation/validation of an algorithm before moving to actual implementation on a specific sensor platform. Something similar to a virtual test before practise.

4.2.1.1. Structure

Castalia is using OMNeT++, so it is suggested that someone has reasonable knowledge of the basic concepts of OMNeT, especially if he wants to use Castalia in an entry-level way (for instance., without using his own protocols/applications).

As per [38,]"OMNeT's basic concepts are modules and messages. A simple module is the basic unit of execution. It accepts messages from other modules or itself, and according to the message, it executes a piece of code. The code can keep state that is altered when messages are received and can send (or schedule) new messages. There are also composite modules. A composite module is just a construction of simple and/or other composite modules".

Castalia's basic module structure is shown in the following diagram:

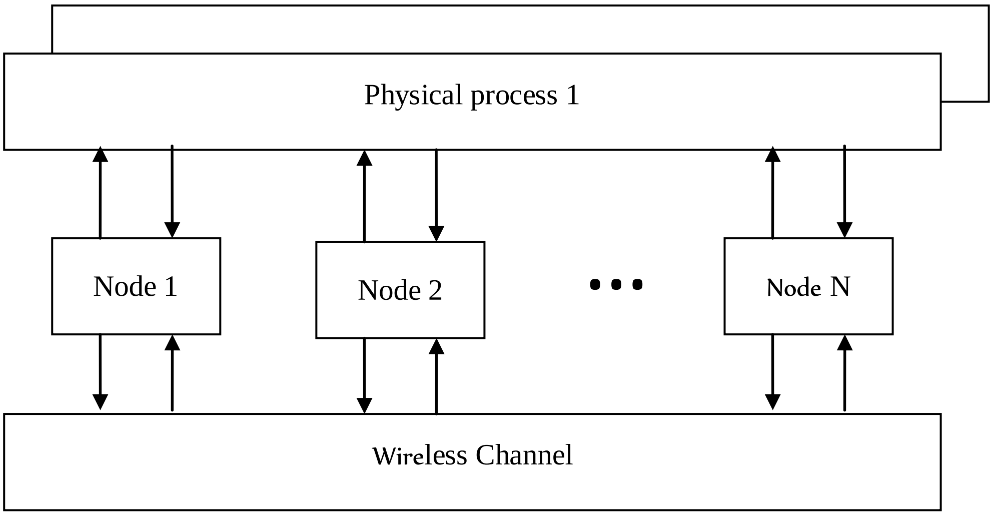


Figure 6 - Castalia simulator's basic module structure as seen in Athanassios Boulis's: "*Castalia. A simulator for Wireless Sensor Networks and Body Area Networks, User's Manual*", version 3.3, p.7,§2.1, NICTA, May 2013

It is easy to figure out that nodes do not directly connect to each other, but instead through the wireless channel. Arrows indicate message exchanging from one module to a different one. When a node is ready to send a packet, this goes through the wireless channel which in turn makes the decision of which nodes should be the receivers. The receiving nodes are also linked through the physical processes as 'listeners' (they monitor the channel). For each and every physical process there is one module which holds the "truth" on the quantity the physical process is representing. The nodes try out the physical process in space and time (by sending a message to the corresponding module) to get their sensor readings back. There can be multiple physical processes, representing the multiple sensing devices a node has making it more realistic.

The node module is a composite one. Next figure shows its internal structure module. Firm arrows mean message sending and the dashed ones signify simple function calling. For instance, most of the modules call a function of the Resource Manager to alert that energy has been consumed. The Application module is the one that users will most commonly edit, by creating a new module to test a new algorithm.

"Communications: MAC and Routing modules, as well as the Mobility Manager module, are also good candidates for change by the user, again usually by creating a new module to implement a new protocol or mobility pattern. Castalia offers support for building our own protocols, or applications by defining appropriate abstract classes. All existing modules are highly tuneable by many parameters" as better described in [38].

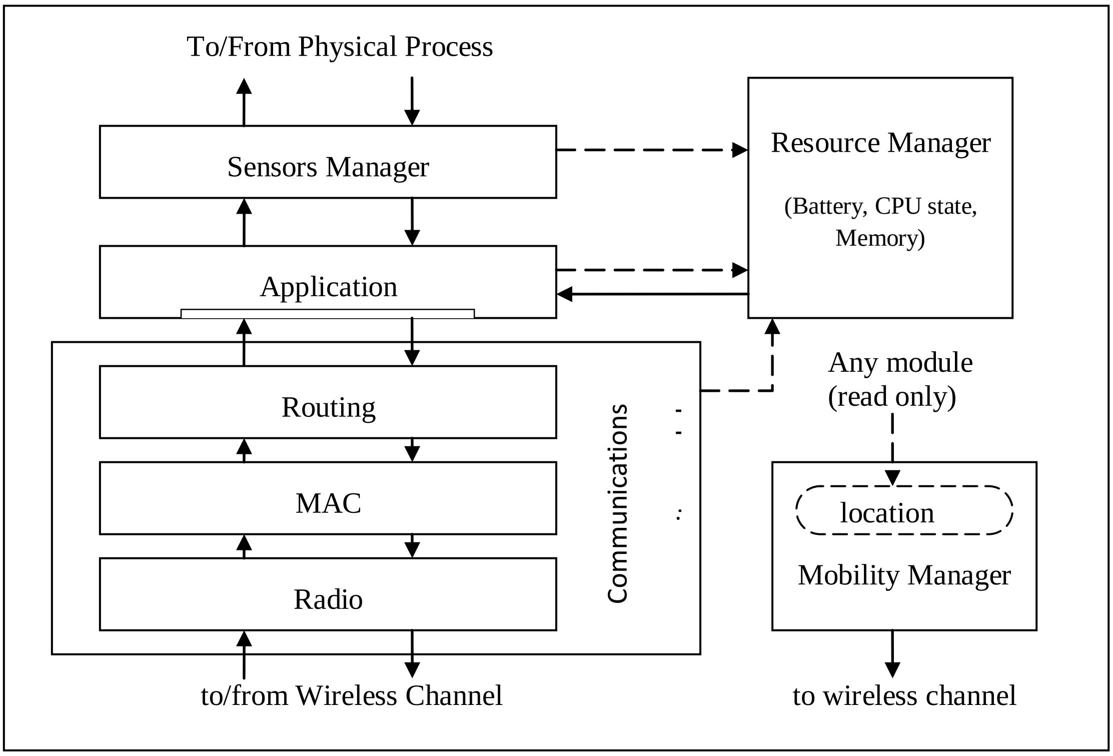


Figure 7 - The node composite module as seen in Athanassios Boulis's: "*Castalia. A simulator for Wireless Sensor Networks and Body Area Networks, User's Manual*", version 3.3, p.8,§2.1, NICTA, May 2013

The structure shown in the above diagram and described in this chapter, is implemented in Castalia with the use of the OMNeT++ NED language. Using this language someone can easily specify modules, i.e., define a module name, module parameters and module interfaces (gates in/out). Secondly, he could define a possible submodule structure (if this is a composite module). Files with the suffix ".ned" contain NED language code. Castalia structure is also depicted in the hierarchy of directories in the source code. Every module corresponds to a directory that always contains a .ned file which defines the module. If the module is multi-part, then there are subdirectories to define the submodules. If it is a simple module then there is C++ code (.cc, .h files) to define its behavior. This exact previous mentioned complete hierarchy of .ned files, defines the overall structure of Castalia simulator. Most likely, the user will not alter these files. Nevertheless, these files are dynamically loaded and processed (using a feature of OMNeT) so that any change does not require the recompilation of Castalia (unless new simple modules with new functions occur).

4.2.1.2. Installation

As previously mentioned, Castalia is based on OMNeT++. The latest version (Castalia 3.3) works fine with OMNeT versions 4.3 to 4.6 . On the contrary, the latest versions of OMNeT (5.x) are not compatible with Castalia.

A Linux or Mac OS X system is recommended. Castalia has been tested with Ubuntu 14 and 16 and OS X 10.9. It is noticeable that Castalia is designed as a command line tool (CLI Environment). Even though many have adapted it to work with the OMNeT graphical IDE (GUI), it is not recommended (nor supported) to be used with it. The instructions given from this point on, refer to a Linux Ubuntu environment.

Installing OMNeT++

OMNeT's Installation guide is for versions 5.x, therefore some things might be different or not needed for versions 4.x. In addition, some basic steps are presented to install OMNeT, assumingly that the gcc compiler and other build tools are already installed in our system (in Ubuntu install build-essential and in Mac install either cmd line developer tools or XCode).

Firstly, download the source code: The following link, [39] works as of March 2017.

If it does not work we can simply go to the OMNeT's website and find out how to download the source code for version 4.6 (or earlier down to 4.3) The zipped source code for version 4.6 is a large file (approximately 188MB) so it might take some time to download. Then we place it in our home directory. Next, we should "Untar" and "unzip" the source file:

```
$ tar xvfz omnetpp-4.6-src.tgz
```

A directory named omnetpp-4.6 will be created. Set environment variables by typing (assuming you are using bash as your shell):

```
$ export PATH=$PATH:~/omnetpp-4.6/bin
$ export LD_LIBRARY_PATH=~/omnetpp-4.6/lib
```

Also add the above two export commands at the end of your .bash_profile file.We are now ready to build OMNeT:

```
$ cd omnetpp-4.6/
$ NO_TCL=1 ./configure
$ make
```

The last command will take a few minutes to complete. We are now done building OMNeT. Castalia does not use the Tcl functionality so we opt to build OMNeT without it. The installation process can be easier if Tcl in not required.

Make sure that OMNeT++ is in the path . For example we can try:

```
$ which opp_makemake
/home/NICTA/aboulis/omnetpp-4.6/bin/opp_makemake
```

Installing Castalia

Initially, we should download the source code from GitHub[40]. If we have downloaded a compressed file (instead of cloning the project) then again we should "untar" and "unzip" it:

```
$ tar –xvzf Castalia-master.tar.gz
```

A new directory will be created, named Castalia/. In there we can find another directory named Castalia, and the User Manual and this installation guide in various forms. We are now ready to build Castalia:

```
$ cd Castalia/Castalia  (or cd Castalia-yourchosenname)
$ ./makemake
```

Wait for a few seconds till the script ends[1]. This automatically generates a Makefile that you can use to build Castalia. Then we type:

```
$ make
```

We wait again for some time until everything is built and check that the soft link CastaliaBin is created in Castalia/Castalia. We have now successfully built Castalia.

4.2.1.3. Using Castalia

A complete Castalia User's Manual was presented from Athanasios Boulis, Version 3.3 NICTA, May 2013, Version 3.3. According to this, in order to run the first simulation we go to the directory Castalia/Simulations/radioTest. It should include one file: omnetpp.ini. This is a configuration file that defines our simulation scenario. Then we can run the input script with no arguments and see what we get:

```
~/Castalia/Simulations/radioTest$ ../../bin/Castalia
```

Executed with no arguments, the script searches the current directory for valid configuration files. If it finds a file, it then parses it and prints the name of the configurations contained in it. In our case it found just one file with five configurations.

```
List of available input files and configurations:
* omnetpp.ini
      General
      InterferenceTest1
      InterferenceTest2
      CSinterruptTest
      varyInterferenceModel
```

If we next run the following:

[1] If the access to the script is refused, make sure the right permissions to the file are granted. If not, we should type chmod u+x makemake and then try again.

```
~/Castalia/Simulations/radioTest$ ../../bin/Castalia -c General
Running Castalia:       Configuration 1/1       Run 1/1           Complete 100%
Time taken 0:00:00.101000
```

We have just run our first simulation! In order to see what’s new in our directory we run the list command ***-ls***:

```
~/Castalia/Simulations/radioTest$ ls
100806-222319.txt  Castalia-Trace.txt  omnetpp.ini
```

4.2.2. Network Simulator 2 (NS-2)

Network Simulator is a discrete event simulator targeted at networking research. It provides substantial support for TCP simulation, routing and multicast protocols over wired and wireless (local and satellite) networks.

It began in 1989 as a variant of the REAL network simulator**[41]** and has evolved substantially over the past few years. In 1995, its development was supported by DARPA through the VINT project**[42]** at LBL, Xerox PARC, UCB, and USC/ISI. Nowadays, its development is supported through DARPA with SAMAN**[43]** and through NSF with CONSER**[44]**, both in collaboration with other researchers including ACIRI**[45]**. Network simulator has always been popular to contributions from other researchers, including wireless code from the UCB Daedelus and CMU Monarch projects and Sun Microsystems. While scientific community has considerable confidence in this particular simulator, it is not a figurative and finished product, but the result of a continuous on-going effort of research and development. In fact, bugs in the software are still being discovered and corrected.

Downloading and building NS

As of November 2005, NS is available at this SourceForge location**[49]**. It requires a moderately up-to-date installation of Tcl/Tk**[50]** (with header files), and two additional packages: tclcl and otcl. Most OS installations do not come with full Tcl/Tk installations or with these other packages, so most likely several packages will be needed to be installed.

Installing NS - Requirements

To build NS, a computer and a C++ compiler is needed. It is developed on several kinds of Unix (FreeBSD, Linux, SunOS, Solaris), but it should better run on an Posix-like computer, possibly with some twisting. NS can also run in Windows (see the dedicated Windows / Cygwin page)**[46]**. Simple scenarios should run on any reasonable machine, but very large scenarios benefit from large amounts of memory. NS is fairly large; the all-in-one package requires about 320MB of disk space to build. If multiple people want to share files in the NS build tree to save space, we may download a simple perl script[47], then follow the instruction in its README. There is detailed instruction[48] from CS599b class of USC. We may also find discussions in the ns-users mailing list archive useful.

4.3. Datasets - Related Work

As far as research is concerned, one of the major questions in scientific society is: What are the different datasets available for network intrusion detection? Some people argue that there is no dataset for network intrusion. That IDS (or other malware detection devices) logs are not the raw data we need. Instead, firewall and router logs are needed and upon these we should apply our own algorithm and data transformation. Their opinion is that IDS logs are the end game, not the working data. However, another approach to that question is that it depends on the IDS problem and our requirements. For instance, in an WBAN environment different categories could be applied:

- The ADFA Intrusion Detection Datasets[51] (2013) are for host-based intrusion detection system (HIDS) evaluation.

- The Public PCAP[52] files for download at NetReSec are a useful resource for PCAP-based evaluation of network-based intrusion detection system (NIDS) evaluation.

- The Cyber Research Center - DataSets - ITOC CDX[53] (2009) dataset provides a comprehensive set of log data under ongoing "sophisticated" attacks.

Furthermore, to achieve machine learning, many of the issues with translating real collected logs of malware and intrusions into an evaluation dataset, solution is labeling and there is no ideal way to do it. Therefore, we probably can't produce a perfect IDS dataset for evaluation. But here are some general, labeled datasets that work towards it:

- The UNB ISCX (2012-..) datasets contain "a range of sophisticated intrusion attacks, botnets and DoS attacks" as Mira Kwak mentions in [54].

- The CSIC[55] 2010 HTTP Dataset in CSV Format (for Weka Analysis) (2010) dataset is from a web penetration testing area for anomaly detection training.

- The Attack Challenge - ECML/PKDD[56] Workshop (2007) dataset contains web penetration testing data.

- Among the direct reconsidering of the KDDCup98 log PCAPs, (for instance those intended to replace the DARPA KDDCup99 dataset for IDS), have been the NSL-KDD[57] Data Set (2007) and gureKddcup[58] data base (2008).

As we can easily understand there are numerous datasets available, but many of them have limitations and are used in different cases. Famous ones are (still until today) DARPA 98/99 and KDD99, despite they have several shortcomings and have been criticized a lot, in example, by Mahoney and Chan[59]. Even so, they are still used, but results and evaluations done by these datasets are questionable. Some improvements had been done by NSL-KDD as already mentioned; Qian et al.[27] lastly, presented another redesign of the DARPA set[60].

Additionally, real-world data, therefore ideally for WBANs, can be found at the MAWI Working Group Traffic Archive and the WIDE project[61]. While real-world (recorded) data is a good challenge for a realistic comparison of IDSs, it often suffers from a missing fundamental truth: their size is over seizing most simulator capabilities. Further packet-, flow- and http- traces can be found at MOME (Cluster of European Projects aimed at Monitoring and Measurement[62], or Consortium Internet 2 (the public link seems to have been removed), CAIDA[63], the Waikato Internet Traffic Storage[64], RIPE[65], the Internet Traffic Archive[66], the UMassTraceRepository[67] or PREDICT[68]. Last but not least, for flow-based systems, labeled datasets can be found at [69].

4.3.1. WSN-DS: A dataset for Intrusion Detection Systems in WBANs

4.3.1.1. Leach Protocol

More appropriate to this thesis topic, a specialized WBAN dataset solution is WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks, as described in [28]. This is a specific dataset for WBANs which detects and classifies four types of Denial of Service (DoS) attacks: Blackhole, Grayhole, Flooding, and Scheduling attacks. It considers the use of LEACH (Low Energy Aware Cluster Hierarchy) protocol which is one of the most popular hierarchical routing protocols in WBANs. This selection was made since LEACH consumes limited energy, therefore increasing the network’s lifetime, and is characterized by its simplicity.

LEACH is considered as a clustering, adaptive, and self organizing protocol. It assumes that Base Station is fixed and located far from sensor nodes. Additionally, all sensor nodes are similar with each other and have limited energy and memory. Sensors can communicate both between themselves and directly with the BS. The main idea of LEACH protocol, according to [28], is to "organize nodes into clusters to distribute the energy among all nodes in the network. Also, in each cluster there is a node called Cluster Head (CH) which aggregates the data received from sensors within its cluster and forward them to the BS".

Figure 8 shows the node structure in LEACH routing protocol. As described in [28], "each round in LEACH protocol consists mainly of two phases: setup phase and steady-state phase. In the setup phase, clusters are formed, whereas in the steady-state phase, sensed data will be transferred to the sink node".

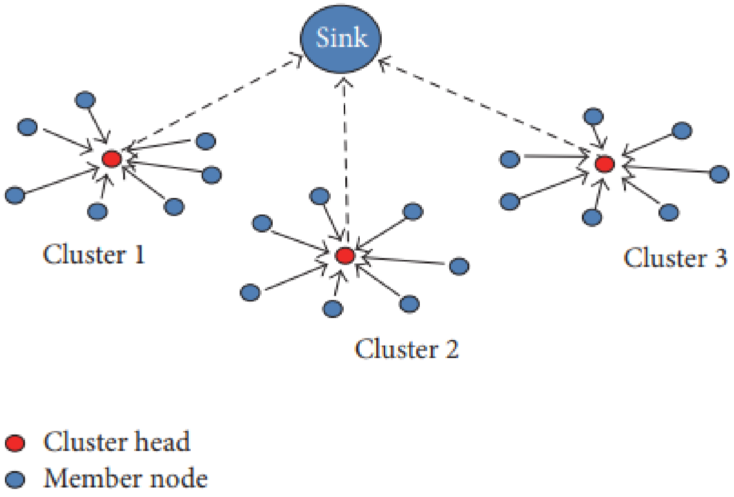


Figure 8 – Nodes Structure in LEACH routing protocol as described in "*WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks*"[28], p2, §2.1.

4.3.1.2. WSN-DS Dataset Description and Creation

In order to create the dataset and collect all the required data from sent and received packets within a WBAN, a service that monitors the network is needed, with minimum cost. On the other hand, it is essential that only required data related to the network which help in detecting, classifying, and then preventing different possible attacks are collected. In order to distribute the load among sensor nodes, each sensor will contribute to the monitoring process and should be able to monitor its neighbor surroundings. The challenge is how to find the suitable number of nodes to be watched by a sensor node in order to monitor all network sensors. Many experiments have been conducted by Iman Almomani et al as shown in [28] to decide on this number, and the summary of the results is shown in Figure 9.

When each sensor node has watched three nodes of its neighbors, it has been observed that the largest number of sensor nodes which could be monitored by a single node was seven. In other words, the BS has received seven different reports about the same node from seven different watching nodes. To ensure that the received information is valid, these reports could be checked for consistency. In some scenarios, few sensor nodes were not monitored by any sensor. This indicates that monitoring three neighboring nodes is inadequate to get information about all network sensor nodes. Additionally, an improvement has occurred when four neighbors are being watched. But only when we increase the amount to five, all sensor nodes are being watched in all five scenarios. Similar results have been obtained when a sensor node was watching six of its neighbors. Consequently, it has been concluded that monitoring five neighbors is enough to get information about all nodes in the network and there is no need to increase the computational complexity by going deeper.

Choosing five neighbors to be monitored is done at the beginning of the simulation. All nodes broadcast a Hello message. Accordingly, each node selects the first five nodes it got it from. Then it interacts with them over the simulation period, so that each node sends a report to its CH at the end of each round. Then the CH sends the received reports to the BS.

4.3.1.3. Attack Models

Four types of DoS attacks in LEACH protocol were implemented in the constructed dataset; Blackhole, Grayhole, Flooding, and Scheduling attacks. This section points out each one of these attacks. To achieve proper distribution of the attacker nodes, the network field has been divided into ten regions. Then the attackers' ratios, were distributed randomly within these regions. According to Iman Almomani et al in [28] the four types are:

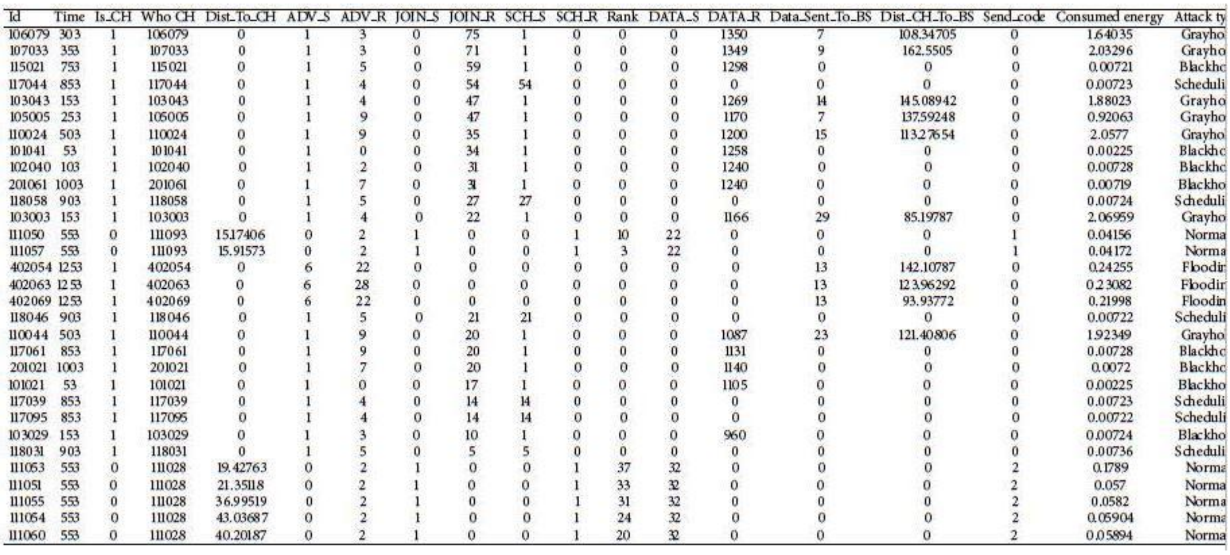
- "Blackhole: To implement this attack in the simulation environment, several attackers' intensities (10%, 30%, and 50%) have been injected randomly to perform the Blackhole attack. These attackers which act as CHs, will drop all the packets relayed through them in their way to the BS.

- Grayhole: Similar to Blackhole attack, 10%, 30%, and 50% of the sensor nodes are injected randomly to implement the Grayhole attack. The decision whether to forward a specific packet or not, is also devised randomly. But the decision can be done selectively based on the sensitivity of the sensed data carried by the packet.

- Flooding attack: This kind of attack has been implemented in several ways in the simulation environment. In some experiments 10 ADV_CH messages were sent by the attacker; other scenarios consider 50 ADV_CH messages to be sent or a random number between 10 and 50. The idea is when more ADV CH messages are sent, more messages will be received and more energy will be consumed.

- Scheduling Attack: The implementation of Scheduling attack is performed by setting the same time for all Cluster Members to send their data packets. Other scenarios assign every two nodes the same time or every five nodes the same time."

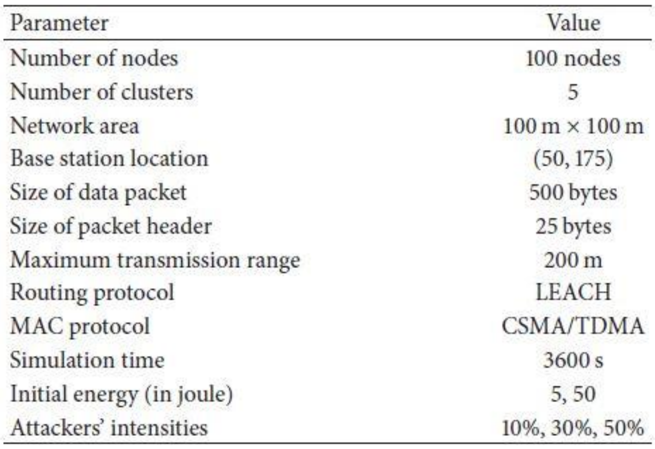
At this point, we should highlight the importance of studying normal and anomaly (attack senario) behaviors of WBAN protocols and their presentation through this specialized dataset. WSN-DS allows several intelligent and data mining approaches to be applied for the purpose of better detection and classification of DoS attacks. As a result, sensor nodes will be more experienced with the normal behaviors and attackers' signatures, and will be able to make correct decisions at the right time.



Id	Time	Is_CH	Who CH	Dist_To_CH	ADV_S	ADV_R	JOIN_S	JOIN_R	SCH_S	SCH_R	Rank	DATA_S	DATA_R	Data_Sent_To_BS	Dist_CH_To_BS	Send_code	Consumed energy	Attack ty
106079	303	1	106079	0	1	3	0	75	1	0	0	0	1350	7	108.34705	0	1.64035	Grayho
107033	353	1	107033	0	1	3	0	71	1	0	0	0	1349	9	162.5505	0	2.03296	Grayho
115021	753	1	115021	0	1	5	0	59	1	0	0	0	1298	0	0	0	0.00721	Blackho
117044	853	1	117044	0	1	4	0	54	54	0	0	0	0	0	0	0	0.00723	Scheduli
103043	153	1	103043	0	1	4	0	47	1	0	0	0	1269	14	145.08942	0	1.88023	Grayho
105005	253	1	105005	0	1	9	0	47	1	0	0	0	1170	7	137.59248	0	0.92063	Grayho
110024	503	1	110024	0	1	9	0	35	1	0	0	0	1200	15	113.27654	0	2.0577	Grayho
101041	53	1	101041	0	1	0	0	34	1	0	0	0	1258	0	0	0	0.00225	Blackho
102040	103	1	102040	0	1	2	0	31	1	0	0	0	1240	0	0	0	0.00728	Blackho
201061	1003	1	201061	0	1	7	0	31	1	0	0	0	1240	0	0	0	0.00719	Blackho
118058	903	1	118058	0	1	5	0	27	27	0	0	0	0	0	0	0	0.00724	Scheduli
103003	153	1	103003	0	1	4	0	22	1	0	0	0	1166	29	85.19787	0	2.06959	Grayho
111050	553	0	111093	15.17406	0	2	1	0	0	1	10	22	0	0	0	1	0.04156	Norma
111057	553	0	111093	15.91573	0	2	1	0	0	1	3	22	0	0	0	1	0.04172	Norma
402054	1253	1	402054	0	6	22	0	0	0	0	0	0	0	13	142.10787	0	0.24255	Floodi
402063	1253	1	402063	0	6	28	0	0	0	0	0	0	0	13	123.96292	0	0.23082	Floodi
402069	1253	1	402069	0	6	22	0	0	0	0	0	0	0	13	93.93772	0	0.21998	Floodin
118046	903	1	118046	0	1	5	0	21	21	0	0	0	0	0	0	0	0.00722	Scheduli
110044	503	1	110044	0	1	9	0	20	1	0	0	0	1087	23	121.40806	0	1.92349	Grayho
117061	853	1	117061	0	1	9	0	20	1	0	0	0	1131	0	0	0	0.00728	Blackho
201021	1003	1	201021	0	1	7	0	20	1	0	0	0	1140	0	0	0	0.0072	Blackho
101021	53	1	101021	0	1	0	0	17	1	0	0	0	1105	0	0	0	0.00225	Blackho
117039	853	1	117039	0	1	4	0	14	14	0	0	0	0	0	0	0	0.00723	Scheduli
117095	853	1	117095	0	1	4	0	14	14	0	0	0	0	0	0	0	0.00722	Scheduli
103029	153	1	103029	0	1	3	0	10	1	0	0	0	960	0	0	0	0.00724	Blackho
118031	903	1	118031	0	1	5	0	5	5	0	0	0	0	0	0	0	0.00736	Scheduli
111053	553	0	111028	19.42763	0	2	1	0	0	1	37	32	0	0	0	2	0.1789	Norma
111051	553	0	111028	21.35118	0	2	1	0	0	1	33	32	0	0	0	2	0.057	Norma
111055	553	0	111028	36.99519	0	2	1	0	0	1	31	32	0	0	0	2	0.0582	Norma
111054	553	0	111028	43.03687	0	2	1	0	0	1	24	32	0	0	0	2	0.05904	Norma
111060	553	0	111028	40.20187	0	2	1	0	0	1	20	32	0	0	0	2	0.05894	Norma

Figure 9 - Sample from WSN-DS Dataset as described in *"WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks"* [28], p.10

In order to gather the required data, NS-2[32] was used in WSN-DS report,[28]. Simulation parameters are summarized in Figure 10. According to the authors, "because different performance metrics are appropriate in different settings, seven performance metrics were used: True Positive Rate (TPR), True Negative Rate (TNR), False Positive Rate (FPR), False Negative Rate (FNR), Overall Accuracy (A), Precision (P), and Root Mean Square Error (RMSE)".



Parameter	Value
Number of nodes	100 nodes
Number of clusters	5
Network area	100 m × 100 m
Base station location	(50, 175)
Size of data packet	500 bytes
Size of packet header	25 bytes
Maximum transmission range	200 m
Routing protocol	LEACH
MAC protocol	CSMA/TDMA
Simulation time	3600 s
Initial energy (in joule)	5, 50
Attackers' intensities	10%, 30%, 50%

Figure 10 - NS-2 simulation parameters seen from *"WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks"* [28], p11

TPR depicts the rate of attack cases identified correctly, TNR represents the rate of normal (no-attack) cases identified correctly, FPR stands for the rate of no-attack cases identified as attacks by the system, and FNR shows the rate of attack cases identified as normal ones. In conclusion, according to the results obtained from applying ANN to WSN-DS dataset, high-level accuracy was achieved in the task of classifying four DoS attacks, to determine whether the protocol is in its normal mode or exposed to any type of attack.

5. CONCLUSIONS

The difficulties of security and intrusion detection in sensor networks, mainly stem from the constraints imposed by the simplicity of sensor devices: limited power, limited communication bandwidth and processing capabilities, and small storage capacity. In this thesis, emphasis was given on detecting the attacker when the prevention measures fail to succeed. Intrusion Detection can complement the intrusion prevention techniques to secure the network. However, new techniques must be developed to make intrusion detection work efficiently for wireless body area networks. Several arguments that such techniques should be distributed and cooperative were deployed. If such a scheme is followed, there will be some interesting findings. The problem was modeled as a multi-objective genetic algorithm optimization issue, to manage the trade-offs among detection accuracy, false positives and resource consumption in WBANs. The proposed detection system is evaluated against the implementation of different types of attacks in WBANs and the performance effectiveness of the detection algorithm was presented through experimental results.

The main goal of this work was to review intelligent intrusion detection and prevention mechanisms, that could work efficiently to limit DoS attacks with reasonable cost in terms of processing and energy. For instance, an evolutionary algorithm (iDetect) for IDS's was examined; along with the way how such algorithms can be leveraged into WBANs in order to make them an intelligent and autonomous network. To achieve this goal, a specialized dataset for WBANs was analyzed, in order to classify four types of DoS attacks. Two WBAN simulators (Castalia and NS-2) were consequently reviewed, but data were collected using NS-2. In addition to including normal behavior, it has also been able for 374.661 records containing the signatures of these four attacks, to be collected. This dataset containing normal and malicious network traffic, was used to obtain the presented experimental results. Additionally, mathematical validation of the created dataset has been provided to ensure its correctness. The specific dataset is called WSN-DS. From the results shown, it can be concluded that ANN trained using WSN-DS dataset, is very useful in classifying DoS attacks, as it was able to achieve high classification accuracy in the presence of more than one attacks, making it a significantly trustworthy and highly suitable solution.

B. ANNEX

References

This Thesis is mainly based on the following references:

[1]. K. Anandumar, C. Jayaumar, K. Arun, M. Sushma, and R. Vikraman, “Intrusion detection and prevention of node replication attacks in wireless body area sensor networks,” International Journal of UbiComp, IJU, vol. 3, no. 3, Jul. 2012.

[2]. G. Thamilarasu, “iDetect: An Intelligent Intrusion Detection System for Wireless Body Area Networks,” Int. J. Secur. Netw., vol. 11, no. 1/2, p. 82–93, March 2016.

[3]. Osman Salem, Yaning Liu, Ahmed Mehaoua, and Raouf Boutaba, Fellow IEEE, "Online Anomaly Detection in Wireless Body Area Networks for Reliable Healthcare Monitoring", IEEE Journal of Biomedical and Health Informatics, Vol. 18, No. 5, September 2014.

[4]. Steve Warren, Jeffrey Lebak, Jianchu Yao, Jonathan Creekmore, Aleksandar Milenkovic, and Emil Jovanov, "Interoperability and Security in Wireless Body Area Network Infrastructures", Proceedings of the 2005 IEEE Engineering in Medicine and Biology 27th Annual Conference Shanghai, China, September 1-4, 2005.

[5]. Chris Otto, Aleksandar Milenkovic, Corey Sanders, Emil Jovanov, "System Architecture of a Wireless Body Area Sensor Network for ubiquitous Health Monitoring", University of Alabama in Huntsville, Journal of Mobile Multimedia, Vol. 1, No.4, p307-326, 2006.

[6]. Keshav Goyal, Nidhi Gupta, Keshawanand Singh, " A Survey on Intrusion Detection in Wireless Sensor Networks" , M.Tech Scholars, School of Computer Science & Engineering, Galgotias University, Greater Noida, International Journal of Scientific Research Engineering & Technology (IJSRET), Volume 2 Issue 2 p 113-126, May 2013.

[7]. National Research Council: "For the Record: Protecting Electronic Health Information" Washington, DC: The National Academies Press, 0309056977, 1997.

[8]. J. Newsome, E. Shi, D. Song, and A. Perrig: "The Sybil attack in sensor networks: Analysis & defenses". In "IPSN '04: Proceedings of the third international symposium on Information processing in sensor networks", p. 259{268}. ACM, New York, USA, 2004.

[9]. Y. Zhang, W. Lee, and Y.-A. Huang: "Intrusion detection techniques for mobile wireless networks" Wireless Networks, vol. 9(5):p. 545{556}, 2003.

[10]. P. Albers, O. Camp, J.-M. Percher, B. Jouga, L. Me, 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), p.1{12}, April 2002.

[11]. O. Kachirski and R. Guha: "Effective intrusion detection using multiple sensors in wireless ad hoc networks". In Proceedings of the 36th Hawaii International Conference on System Sciences (HICSS '03), p. 57, January 2003.

[12]. A. Siraj, S. Bridges, and R. Vaughn: "Fuzzy cognitive maps for decision support in an intelligent intrusion detection system". In IFSA World Congress and 20th North American Fuzzy Information Processing Society (NAFIPS) International Conference, vol.4, p. 2165{2170}, July2001.

[13]. S. Marti, T. J. Giuli, K. Lai, and M. Baker: "Mitigating routing misbehavior in mobile ad hoc networks". In Proceedings of the 6th International Conference on Mobile Computing and Networking (MobiCom'00), p. 255{265}, August 2000.

[14]. O. Kachirski and R. Guha: "Intrusion detection using mobile agents in wireless ad hoc networks". In KMN '02: Proceedings of the IEEE Workshop on Knowledge Media Networking, p. 153, 2002.

[15]. A. P. da Silva, M. Martins, B. Rocha, A. Loureiro, L. Ruiz, and H. C.Wong: "Decentralized intrusion detection in wireless sensor networks". In Proceedings of the 1st ACM International Workshop on Quality of Service & Security in Wireless and Mobile Networks (Q2SWinet '05),p.16{23}, ACM Press, October 2005.

[16]. I. Onat and A. Miri: "An intrusion detection system for wireless sensor networks". In Proceeding of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, vol. 3, p. 253{259}, Montreal, Canada, August 2005.

[17]. C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami: "Intrusion detection for routing attacks in sensor networks". International Journal of Distributed Sensor Networks, 2005.

[18]. V. Bhuse and A. Gupta: "Anomaly intrusion detection in wireless sensor networks". Journal of High Speed Networks, vol. 15(1), p. 33{51}, 2006.

[19]. R. Roman, J. Zhou, and J. Lopez: "Applying intrusion detection systems to wireless sensor networks". In Proceedings of IEEE Consumer Communications and Networking Conference (CCNC '06), p. 640{644}, Las Vegas, USA, January 2006.

[20]. F. Anjum D. Subhadrabandhu, S. Sarkar, and R. Shetty: "On optimal placement of intrusion detection modules in sensor networks". In BROADNETS '04: Proceedings of the First International Conference on Broadband Networks (BROADNETS'04), p. 690{699}, 2004.

[21]. Movassaghi S., Abolhasan M., Lipman J., Smith, D. and Jamalipour A: "Wireless body area networks: a survey", IEEE Communications Surveys and Tutorials, Vol. 16, p.1658, 2014.

[22]. Malan D., Fulford-Jones T., Welsh M. and Moulton S.: "Codeblue: an ad hoc sensor network infrastructure for emergency medical care", International Workshop on Wearable and Implantable Body Sensor Networks, Imperial College London, UK, 2004.

[23]. Virone G., Wood A., Selavo L., Cao Q., Fang L., Doan T., He Z., Stoleru R., Lin S. and Stankovic J: "An assisted living oriented information system based on a residential wireless sensor network", 1st Transdisciplinary Conference on Distributed Diagnosis and Home Healthcare, D2H2 Arlington, VA, p.95–100, 2006.

[24]. Malasri K. and Wang L: "Snap: an architecture for secure medical sensor network", 2nd IEEE Workshop on Wireless Mesh Networks, WiMesh, p.160–162, 2006.

[25]. Xu F., Qin Z., Tan C.C, Wang B. and Li Q: "IMD Guard: securing implantable medical devices with the external wearable guardian", INFOCOM IEEE, p.1862–1870, 2011.

[26]. Lee C.H, Shin S.W. and Chung J.W: "Network intrusion detection through genetic feature selection", 7th ACIS International Conference on Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing, 2006. SNPD 2006,NV Las Vegas,US, p.109–114, 2006.

[27]. Qian J, Xu C, Shi M.: "Redesign and Implementation of Evaluation Dataset for Intrusion Detection System", In: Müller G. (eds) Emerging Trends in Information and Communication Security, ETRICS, Lecture Notes in Computer Science, vol 3995, Springer Berlin, Heidelberg, 2006.

[28]. Iman Almomani, Bassam Al-Kasasbeh and Mousa AL-Akhras: "WSN-DS: A Dataset for Intrusion Detection Systems in Wireless Sensor Networks", Journal of Sensors (2):1-16, January 2016.

[29]. Asfaw B, Bekele D, Eshete B, Villafiorita A. and Weldemariam K: "Host-based anomaly detection for pervasive medical systems", CRiSIS IEEE, Montreal QC, Canada, p.1–8, 2010.

[30]. Coppolino L. and Romano L. : "Open issues in IDS design for wireless biomedical sensor networks", Intelligent Interactive Multimedia Systems and Services, Vol. 6, SpringerLink, p.231–240, 2010.

[31]. Zhang M, Raghunathan A. and Jha N.K: "Medmon: securing medical devices through wireless monitoring and anomaly detection:, IEEE Trans. Biomed. Circuits and Systems, Vol. 7, No. 6, p.871–881, 2013.

[32]. The Network Simulator—ns-2, http://www.isi.edu/nsnam/ns/

[33]. J. Xu, J. Wang, S. Xie, W. Chen, and J.-U. Kim: “Study on intrusion detection policy for wireless sensor networks,” International Journal of Security and Its Applications, vol. 7, no. 1, p.1–6, 2013.

[34] S.Khan andK.-K. Loo: “Real-time cross-layer design for a large scale flood detection and attack trace-back mechanism in IEEE 802.11 wireless mesh networks,” Network Security, vol. 2009, no.5, p. 9–16, 2009.

[35] N. A. Alrajeh, S. Khan, and B. Shams: “Intrusion detection systems in wireless sensor networks: a review,” International Journal of Distributed Sensor Networks, vol. 9, no. 5,p. 1–7, 2013.

[36] A. Abid, A. Kachouri, and A. Mahfoudhi: “Anomaly detection in WSN: critical study with new vision,” in Proceedings of the International Conference on Automation, Control, Engineering and Computer Science (ACECS ’14), p. 37–46, 2014.

[37] H. Jadidoleslamy: “A high-level architecture for intrusion detection on heterogeneous wireless sensor networks: hierarchical, scalable and dynamic reconfigurable, ”Wireless Sensor Network,vol. 3, no. 7, p. 241–261, 2011.

[38] Athanassios Boulis: "Castalia - A simulator for Wireless Sensor Networks and Body Area Networks, Version 3.3,User's Manual", NICTA, May 2013

Links

[38] Omnet++link: http://www.omnetpp.org/
[39] OMNeT++ 4.6 (source + IDE, tgz):https://omnetpp.org/component/jdownloads/send/32-release-older-versions/2290-omnet-4-6-source-ide-tgz
[40] Castalia source code: https://github.com/boulis/Castalia
[41] Real NS : http://www.cs.cornell.edu/home/skeshav/real/overview.html
[42] Vint Project : http://www.isi.edu/nsnam/vint/index.html
[43] Saman : http://www.isi.edu/saman/index.html
[44] Conser : http://www.isi.edu/conser/index.html
[45] Aciri : http://www.aciri.org/
[46] Win/Cygwin page:
http://nsnam.isi.edu/nsnam/index.php/Running_Ns_and_Nam_Under_Windows_9x/2000/XP_Using_Cygwin
[47] Simple Perl script : http://www.isi.edu/nsnam/dist/dup.tar.gz
[48] Detailed instruction: http://netweb.usc.edu/cs599sp00/sharedns.html
[49] Source Forge : http://sourceforge.net/projects/nsnam/
[50] Tcl/tl : http://www.tcl.tk/
[51] Adfa : http://www.cybersecurity.unsw.adfa.edu.au/ADFA%20IDS%20Datasets/
[52] Pcap : http://www.netresec.com/?page=PcapFiles
[53] Itoc : http://www.westpoint.edu/crc/SitePages/DataSets.aspx
[54] UNB : http://www.unb.ca/research/iscx/dataset/index.html
[55] Csic : http://bit.ly/csic-2010-http-dataset_csv
[56] ECML : http://www2.lirmm.fr/pkdd2007-challenge/#dataset
[57] NSC-KPD : http://nsl.cs.unb.ca/NSL-KDD/
[58] Gurekddcup : http://www.sc.ehu.es/acwaldap/gureKddcup/gureKddcup_index.htm
[59] Springer:
https://www.researchgate.net/deref/http%3A%2F%2Flink.springer.com%2Fchapter%2F10.1007%2F978-3-540-45248-5_13
[60] DARPA:
https://www.researchgate.net/deref/http%3A%2F%2Flink.springer.com%2Fchapter%2F10.1007%252F11766155_32
[61] MAWI : http://mawi.wide.ad.jp/mawi/
[62] MOME : http://www.ist-mome.org/database/MeasurementData/index8db7.html
[63] CAIDA : http://www.caida.org/data/overview
[64] Wand : http://wand.net.nz/wits/
[65] RIPE : https://labs.ripe.net/datarepository/data-sets
[66] ITA : http://ita.ee.lbl.gov/html/traces.html
[67] Umass : http://traces.cs.umass.edu/index.php/Network/Network

[68] Predict : http://www.predict.org/Default.aspx?tabid=175
[69] Simple Web : http://www.simpleweb.org/wiki/Labeled_Dataset_for_Intrusion_Detection