

Security Models for Wireless Sensor Networks

Sophia Kaplantzis

March 20, 2006

Supervisors: Dr N. Mani, Prof. M. Palaniswami, Prof. G. Egan

sophia.kaplantzis@eng.monash.edu.au

Abstract

Wireless Sensor Networks (WSNs) are a new technology foreseen to be used increasingly in the near future due to their data acquisition and data processing abilities. Security for WSNs is an area that needs to be considered in order to protect the functionality of these networks, the data they convey and the location of their members. The security models & protocols used in wired and other networks are not suited to WSNs because of their severe resource constrictions.

In this report we highlight the research to date in the area of security for WSNs and propose a solution based on intrusion detection systems and efficient classifiers. Our hope is to generate a security model that will provide energy efficiency and fault tolerance to WSNs under attack.

Contents

| | | |
|----------|--|-----------|
| 1 | Introduction | 1 |
| 1.1 | Applications | 2 |
| 1.2 | Wireless Sensor Network Model | 2 |
| 1.3 | Hardware Specifications | 3 |
| 1.4 | Protocols | 4 |
| 1.5 | DoS Threats to WSNs | 6 |
| 1.6 | Primitive Countermeasures | 8 |
| 1.7 | Intrusion Detection and Classification | 11 |
| 1.7.1 | Intrusion Detection Techniques | 11 |
| 1.7.2 | Artificial Neural Networks | 11 |
| 1.7.3 | K-means nearest neighbours | 12 |
| 1.7.4 | Support Vector Machines | 13 |
| 1.7.5 | Hidden Markov models | 14 |
| 1.7.6 | Graph Theory | 15 |
| 1.7.7 | Game Theory | 16 |
| 1.7.8 | Honeypots | 16 |
| 2 | Research Aims | 17 |
| 3 | Literature Review | 17 |
| 3.1 | Encryption | 18 |
| 3.1.1 | Shared Keys | 18 |
| 3.1.2 | Secure Groups | 18 |
| 3.2 | Data Aggregation | 19 |
| 3.3 | Secure Protocols | 19 |
| 3.4 | Intrusion Detection Systems | 19 |
| 3.5 | Privacy | 20 |
| 3.6 | Other Issues | 21 |
| 3.7 | Open Research Areas | 21 |
| 4 | Preliminary Research | 22 |
| 4.1 | Inadequacy of Available Simulation Tools | 22 |
| 4.2 | Development of a Preliminary Simulator | 23 |
| 4.3 | Preliminary SVM classification | 23 |
| 4.4 | Hardware Configuration | 24 |
| 5 | Proposed Research | 25 |
| 5.1 | Research Plan | 26 |
| 5.1.1 | The Hybrid Intrusion Detection System | 26 |
| 5.1.2 | Attack Replication/Verification | 26 |
| 5.1.3 | Classification | 29 |
| 5.1.4 | Recovery | 30 |
| 5.1.5 | Benchmarking | 30 |
| 5.2 | Timetable | 30 |
| 5.3 | Resources | 31 |
| 6 | Conclusions | 32 |

List of Figures

| | | |
|----|---|----|
| 1 | Wireless Sensor Nodes and Networks | 1 |
| 2 | Sensor Network Communication Structure | 3 |
| 3 | Sensor Node Hardware Components | 4 |
| 4 | Sensor Node Protocol Stack | 5 |
| 5 | Hello Flood Attack | 8 |
| 6 | Defense Against Jamming Attacks | 9 |
| 7 | An Artificial Neural Network | 12 |
| 8 | K-means clustering | 13 |
| 9 | Hyperplane of a Support Vector Machine | 14 |
| 10 | State Transitions of a Hidden Markov Model | 15 |
| 11 | A network Represented as a Graph | 15 |
| 12 | Winnie-the-Pooh Lured by the Honey | 17 |
| 13 | Screenshot from the Preliminary Simulator | 24 |
| 14 | Research Overview | 25 |
| 15 | Ambient Systems SmartTag Technology | 27 |
| 16 | Possible sources of data for network analysis | 28 |
| 17 | Gannt chart for proposed research | 31 |

List of Tables

| | | |
|---|---|---|
| 1 | Mica vs μ node components | 4 |
|---|---|---|

1 Introduction

The miniturisation of electronics, along with the advances in wireless communications and the development of multi-functional sensors, has led to the birth of a new technology named Wireless Sensor Networks. The magazine Business Week [1] even went as far as to identify micro-sensor networking as one of the 21 most important technologies of the 21st century.

A wireless sensor network is simply defined as a large collection of sensor nodes, each equipped with its own sensor, processor and radio transceiver. Such networks have substantial data acquisition and data processing capabilities and for this reason are deployed densely throughout the area where they will monitor specific phenomena. However, due to the lack of tamper-resistant packaging and the insecure nature of wireless communication channels, these networks are vulnerable to internal and external attacks!

The main objective of our research is to develop a **fault tolerant** and **energy efficient** intrusion detection model to secure these networks from such malicious attacks.

The remaining transfer report is structured as follows:

- *Section 1* summarises background information associated WSNs, security and classifiers.
- *Section 2* states our research aims.
- *Section 3* is a literature survey in the area of security for WSNs
- *Section 4* outlines the preliminary research to date
- *Section 5* describes future research plans, including timelines and required resources.
- *Section 6* concludes the paper, highlighting the benefits and the novelty of our proposed research.

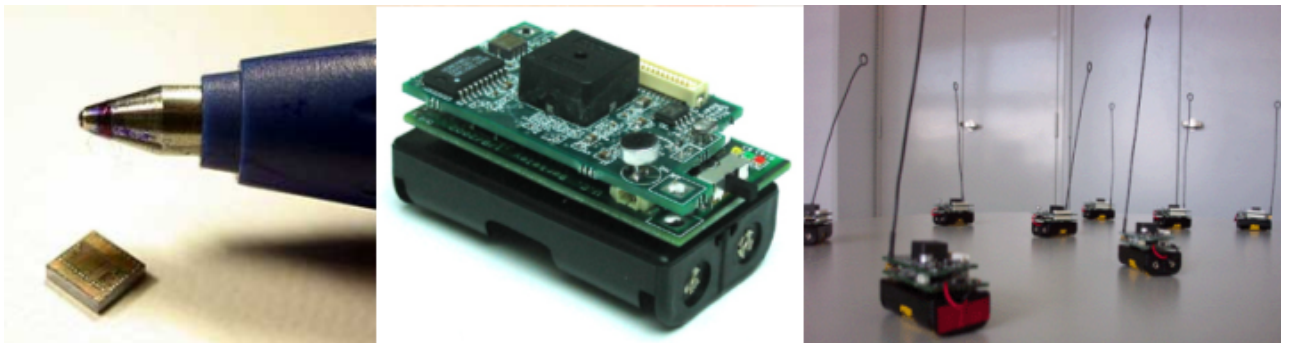


Figure 1: Wireless Sensor Nodes and Networks

1.1 Applications

The appetite for previously unrealisable data gathering along with existence of static infrastructure has led to the creation of application dependant sensor networks. Sensor networks were primarily designed for real-time collection and analysis of low level data in environments. For this reason they are well suited to a multitude of monitoring and surveillance applications. Popular applications include wildlife monitoring, bushfire response, military command, distributed robotics, industrial quality control, observation of critical infrastructures, smart buildings, intelligent communications, traffic monitoring, examining human heart rates etc.

For some sensor network applications however security is crucial as they can be deployed in hostile environments with active intelligent opposition. One obvious example are battlefield applications where there is a pressing need for secrecy of location and resistance to subversion and destruction of the network. Less obvious but just as important security reliant applications include [2] :

- *Disasters* : In many disaster scenarios, especially those induced by terrorist activities, it may be necessary to protect the location of casualties from unauthorised disclosure
- *Public Safety* : In applications where chemical, biological or other environmental threats are monitored, it is vital that the availability of the network is never threatened. Attacks causing false alarms may lead to panic responses or even worse total disregard for the signals.
- *Home Healthcare*: In such applications, privacy protection is essential. Only authorised users should be able to query and monitor the network.

1.2 Wireless Sensor Network Model

Many people confuse WSNs with their closest "ancestors" the ad hoc network (a.k.a wireless multihop networks). The reality is however, that WSNs are unlike adhoc networks in the sense that WSNs are resource limited, they are deployed densely, they are prone to failures, the number of nodes in WSNs is several orders higher than that of ad hoc networks, WSN network topology is constantly changing, WSNs use a broadcast communication mediums and finally sensor nodes don't have a global identification tags [3].

The major components of a typical sensor network are: sensor nodes, the sensor field, the sink and the task manager, as shown in *Figure2*. We proceed to define these components in further detail;

A *sensor field* can be considered as the area in which the nodes are placed i.e. the area in which we expect a particular phenomenon to occur.

Sensors nodes or motes are the heart of the network. They are in charge of collecting data and routing this information back to a *sink*. The exact specification of these sensor node devices is given in Section 1.3.

A *sink* is a sensor node with the specific task of receiving, processing and storing data from the other sensor nodes. They serve to reduce the total number of messages that need to be sent, hence reducing the overall energy requirements of the network. Such points are usually assigned dynamically by the network. Regular nodes can also be considered as sinks if they delay outgoing messages until they have aggregated

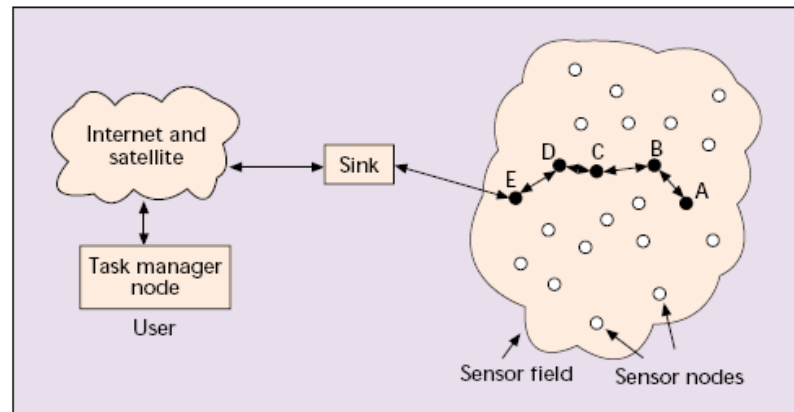


Figure 2: Sensor Network Communication Structure

enough sensed information. For this reason sinks are also known as data aggregation points.

The *task manager* or base station is centralised point of control within the network, which extracts information from the network and disseminates control information back into the network. It also serves as a gateway to other networks, a powerful data processing/storage centre and an access point for a human interface. Hardware-wise the base station is either a laptop or a workstation. Data is streamed to these workstations either via the internet, wireless channels, satellite etc.

So hundreds to several thousands nodes are deployed throughout a sensor field to create a wireless multihop network. Nodes can be deployed as dense as 20 nodes/m^3 . Nodes can use wireless communication media such as infrared, radio, optical media or bluetooth for their communications. The transmission range of the nodes varies according to the communication protocol is use.

Finally WSNs can be described on a higher level as the combination of two different network entities [4];

- *The data acquisition network:* A collection of sensor nodes and the base station. The sensor networks measure physical data and the base station collects the information from the nodes and forwards control data to the network environment.
- *The data dissemination network:* Interfaces the data acquisition network to a user and is a collection of wired and wireless networks.

1.3 Hardware Specifications

Sensor nodes are conventionally made up of four basic components as shown in *Figure3*: a sensing unit, a processing unit, a radio transceiver and a power unit [3]. Additional components may include location finding systems such as GPS, mobilizers (that are required to move the node in specific applications i.e. UAVs) and power generators.

The following is a brief explanation on what each of these components does; The analog signals that are measured by the sensors are digitized via an *ADC* and in turn fed into the *processing unit*. The processing unit and its associated *storage* manage the procedures that make the sensor node carry out its assigned sensing and collaboration

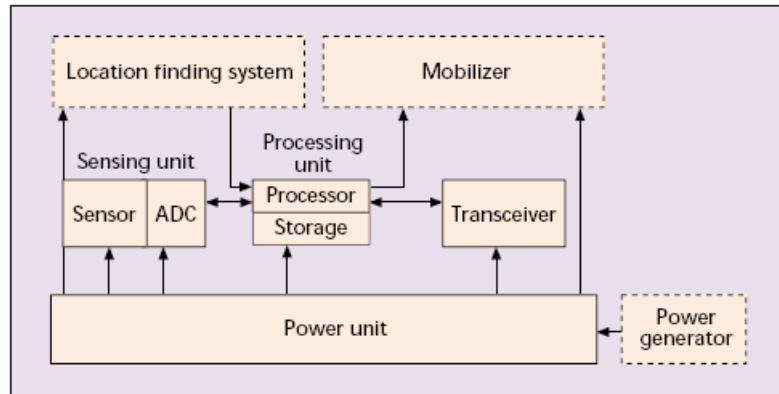


Figure 3: Sensor Node Hardware Components

tasks. The *radio transceiver* connects the node with the network and serves as the communication medium of the node.

The *power unit* is the most important component of the sensor mote because it implicitly determines the lifetime of the entire network. Due to size limitations AA batteries or quartz cells are used as the primary sources of power. To give an indication of the energy consumption involved, the average sensor node will expend approximately 4.8mA receiving a message, 12mA transmitting a packet and $5\mu\text{A}$ sleeping [3]. In addition the CPU uses on average 5.5mA when in active mode. Scavenging units, such as solar cells, may be added to the motes to support the power unit.

Xbow [5] and Ambient Systems [6] are two companies that produce sensor nodes for commercial use. To give an idea of the resource limitations involved, *Table 1* shows specifics of the Xbows best selling Mica mote and Ambient System's leading product the μnode . The Mica mote has been used extensively throughout studies at Berkeley University, in an attempt to develop an operating system for WSNs (TinyOS) and the μnode is the type of node that we will hopefully be conducting our research on.

Table 1: Mica vs μnode components

| | MICAz | μNODE |
|------------------|---------------------------|------------------------------------|
| Processor | 4MHz 8-bit Atmel | 16MHz 16-bit TI MSP430 |
| Memory | 4KB RAM, 512KB flash | 10KB RAM, 1MBit flash |
| Radio | 916MHz, 40Kbps, 35m range | 868/917MHz, 50m range |

Note:- Both motes have sensor boards that allow for mounting of sensors such as magnetometers, accelerometers, microphones, sounders, temperature sensors etc.

1.4 Protocols

Sensor nodes like any other telecommunications device adhere to a specific protocol stack (*Figure 4*). Layered network architectures are adopted because they most certainly always improve the robustness of a system. In this section, we specify the task of each layer of the stack and the most common protocols coupled with each layer [3].

Note that a lot of research is still being conducted in perfecting the protocol stack for sensor network, so the exact protocols are yet to be concreted.

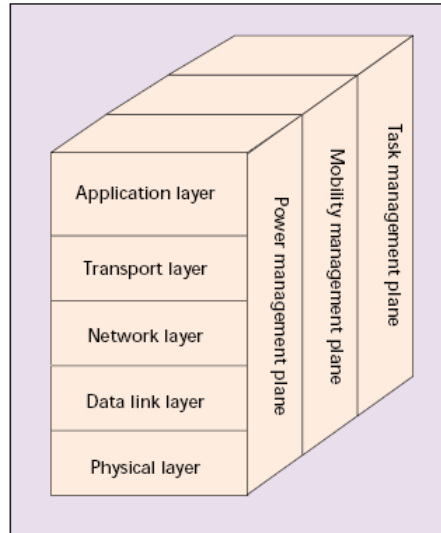


Figure 4: Sensor Node Protocol Stack

1. *The Physical Layer* is responsible for carrier frequency generation, frequency selection, signal detection, modulation and data encryption. Techniques such as Ultra Wideband, Impulse Radio and Pulse Position modulation have been used to reduce complexity and energy requirements, whilst improving reliability and reducing path loss effects and shadowing.
2. *The Data Link Layer* is responsible for medium access, error control, multiplexing of data streams and data frame detection. It ensures reliable point to point and point to multihop connections in the network. Due to the network constraints conventional MAC protocols are not suited to sensor networks. Some widespread data link layer protocols include: SMACS (Self-Organised Medium Access Control for Sensor Networks) [7], EARS (Eavesdrop and Register) [7], CSMA-Based medium Access Protocols [8] and Hybrid TDMA/FDMA-Based protocols [9].
3. *The Network Layer* is responsible for routing information through the sensor network i.e. finding the most efficient path for the packet to travel on its way to a destination. Most protocols can be categorised under one of the following techniques: gossiping, flooding, SMECN (Small Minimum Energy Communication Network) [10], SPIN (Sensor Protocols for Information via Negotiation) [11], SAR (Sequential Assignment Routing) [7], LEACH (Low Energy Adaptive Clustering Hierarchy) [12] and Directed Diffusion [13].
4. *The Transport Layer* is needed when the sensor network intends to be accessed through the internet. However, no scheme has been devised to fully address this issue. Modified TCP/UDP like protocols may be an appropriate solution but this is yet to be established.
5. *The Applications Layer* is responsible presenting all required information to the application and propagating requests from the application layer down to the lower

layers. Some preliminary protocols in this area include SMP [14](Sensor Management Protocol), TADAP (Task Assignment and Data Advertisement Protocol) [14], and SQDDP (Sensor Query and Data Dissemination Protocol) [14].

1.5 DoS Threats to WSNs

Many sensor network deployments are security sensitive and attacks against them provoke the possibility for real-world damage to the health and safety of people. Hardware failures, bugs, resource exhaustion, malicious attacks and environmental conditions can diminish or even eliminate a networks capacity to perform as expected. Such conditions are defined as Denial of Service (DoS) attacks in the literature.

In the previous section we outlined the layered network architecture of sensor networks. In this section we specify DoS vulnerabilities to the first four layers of the stack (*Figure 4*), as specified in studies conducted by Wood and Stankovic [2].

1. *Physical Layer Attacks*: Jamming and tampering are the most common attacks to the physical layer of a WSN.
 - a) Jamming interferes with the radio frequencies the nodes are using. Only a few jamming nodes can put a considerable amount of the nodes out of order. If the adversary can block the entire network then that constitutes complete DoS. However, larger networks are harder to block in their entirety.
 - b) Nodes may fall victims to physical tampering, especially if they are part of a network that covers a vast area. A tampering attacker may damage a sensor, replace the entire node or part of its hardware or even electronically interrogate the nodes to gain access to sensitive information, such as shared cryptographic keys and how to access higher communication layers.
2. *Data Link Layer Attacks*: Collisions, unfairness or exhaustion attacks can be launched against the data link layer of a sensor network.
 - a) Collisions are a type of link layer jamming. If an attacker can corrupt an octet of transmission such that a checksum mismatch occurs, then the entire packet can be disrupted. Corrupted ACK messages usually lead to costly exponential backoff in some MAC protocols. A compromised node may also intentionally deny access to a channel, whilst expending less energy required by full-time jamming of the channel.
 - b) Unfairness is a weaker form DoS that is done by abusing MAC priority schemes. Such an attack usually leads to loss of real-time deadlines and hence degradation of service.
 - c) Exhaustion of battery resources may occur when naive link layer implementations attempt repeated retransmission even after unusually late collisions. A variation of this attack is when a self sacrificing node continuously ask for access to a channel, forcing its neighbours to respond with a clear to send message.
3. *Network Layer Attacks*: Wood and Stankovic [2] inform us that neglect, greed, homing, misdirection, authorisation, probing, blackholes and monitoring are possible routing layer attacks. In a later more detailed study, Karlof and Wagner [15] put specific names and methodologies to these attacks.
 - a) Spoofed, altered or replayed routing information: This is the most direct attack. By spoofing, altering or replaying routing information the attacker can

complicate the network by creating routing loops, attracting or repelling traffic, generating false error messages, shortening or extending source routes or partitioning the network.

b) Selective Forwarding: In such an attack the adversary includes himself/herself in a data flow path of interest. Then the attacker may choose not to forward certain packets and drop them causing a sort of black hole. A variation of this attack is when the adversary only drops packets coming from a specific source whilst reliably forwarding other packets. Such attacks are much harder to detect than black hole attacks.

c) Sinkhole Attacks: The goal of a sinkhole attack is to lure traffic to a malicious part of the network. Such attacks are usually the launching block for other attacks such as selective forwarding. Sinkholes work by making a compromised node attractive to its neighbours. This is done by advertising high quality routes i.e low latency routes. Fooled neighbours will then forward all their data destined to the base station to the lying node. Sensor networks are susceptible to these attacks due to their multihop nature and the specialised communication patterns they use.

d) The Sybil Attack: The Sybil attack targets fault tolerant schemes such as distributed storage, dispersity, multipath routing and topology maintenance. This is done by having a malicious node present multiple identities to the network. This attack is especially confusing to geographic routing protocols as the adversary appears to be in multiple locations at once.

e) Wormholes: In these attacks the adversary tunnels messages received in one part of the network over a low latency link, to another part of the network where the messages are then replayed. The simplest occurrence of this attack is to have a malicious node forwarding data between two legitimate nodes. Wormholes often convince distant nodes that they are neighbours, leading to quick exhaustion of their energy resources. An attacker close to the base station can completely disrupt routing by creating well positioned wormholes that convince nodes multiple hops from the base station that they are only a couple of hops away through the wormhole. When this attack is coupled with selective forwarding and the Sybil attack it is very difficult to detect.

f) Hello flood attacks: In many routing protocols, nodes broadcast hello messages to announce their presence to their neighbours. A node receiving such a message can assume that the node that sent the message is within its range. An attacker with a high powered antenna can convince every node in the network that it is their neighbour. If the attacker also advertises a high quality route it can get every node to forward data to it. Nodes at a large distance from the attacker will be sending their messages into oblivion leaving the network in a state of confusion. This attack can also be thought of as a type of broadcast wormhole. Routing protocols dependant on localised information are extremely vulnerable to such attacks.

g) Acknowledgement Spoofing: Protocols that choose the next hop based on reliability issues are susceptible to acknowledgments spoofing. Here the attacker spoofs acknowledgement convincing the sender that a weak link may be strong or a dead node is alive. This results in packets being lost when traveling along such links.

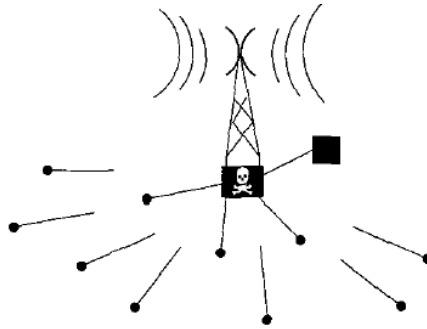


Figure 5: Hello Flood Attack

4. *Transport Layer Attacks*: Finally the transport layer can be attacked via flooding or desynchronisation.

a) The goal of flooding attacks is to exhaust memory resources of a victim system. Similar to TCP SYN attacks the attacker sends many connection establishment requests, forcing the victim to allocate memory in order to maintain the state for each connection.

b) In desynchronisation attacks the hacker forges messages between endpoints. Control flags and sequence numbers are usually modified. If the attacker can get the timing right, he might prevent the endpoints from ever exchanging messages as they will be continually requesting retransmission of previous erroneous messages. This attack leads to an infinite cycle that wastes energy.

To conclude this section we present possible threat models to WSNs based on the tools attackers have at their disposal.

An attacker may have access only to a few nodes which he or she has compromised. Such attacker are classified as *mote class attackers*. Alternatively an attacker may have access to more powerful devices such as laptops, hence the definition *laptop class attacker*. Such attackers have powerful CPUs, great battery power, high power radio transmitter and sensitive antennas at their disposal and pose a much larger threat to the network. For example a few nodes can jam a few radio links whereas a laptop can jam the entire network.

Finally, attacks launched on a network may be *insider* or *outsider* attacks. In outsider attacks the attacker has no special access to the network. In insider attacks however, the attacker is considered to be an authorised participant of the network. Such attacks are either launched from compromised sensor nodes running malicious code or laptops using stolen data (cryptographic keys & code) from legitimate nodes.

1.6 Primitive Countermeasures

Now that we have given a background describing vulnerabilities of sensor networks, we will discuss some possible countermeasures to the above mentioned attacks. These countermeasures are suggested throughout [2] and [15] and we will be addressing them in this section.

Note that these countermeasures are yet to be proven fully effective and haven't been implemented in any of the available software/hardware.

- **Jamming:** There are many deficit solutions to the problem of jamming. Spread spectrum communication may be a temporary countermeasure until the jammers figure out how to follow the hopping sequence or how to jam a wider part of the band. Code spreading similar to that used in mobile phone may solve the problem but requires more design effort, costs and power. Switching to a low power cycle may allow for conservation of energy whilst the network is under attack. Another solution may be to change the mode of communication to Infra Red or optical but this may be costly. In such jamming attacks however it would be ideal if the nodes under attack could alert the rest of the network and ideally the base station about what is going on. Jammed nodes should attempt to inform neighbouring nodes of the attack during jamming gaps and these nodes should in turn inform the base station (*Figure6*). Neighbouring nodes can also assume a jamming attack if they observe change in the neighbouring background noise.

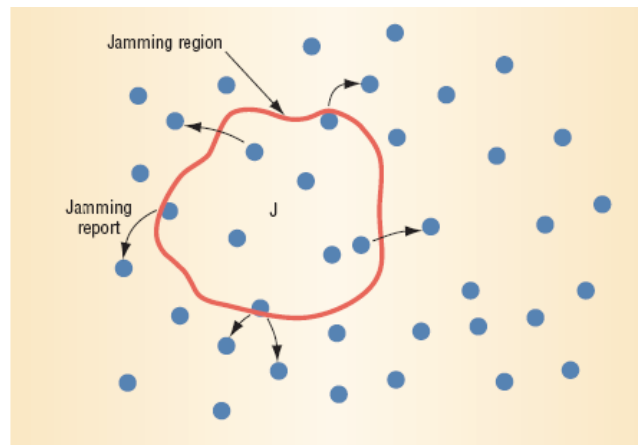


Figure 6: Defense Against Jamming Attacks

- **Tampering:** The ideal solution to tampering is providing tamper resistant packaging for the motes. This however is a very costly procedure which needs to be considered during design time and will no doubt increase the average price of motes. Camouflaging is another option. The motes may also be programmed to erase sensitive data upon capture.
- **Collision:** Collision detection is the obvious solution here, however it hasn't yet been proved to be completely effective. One may choose to employ error correcting codes, however these can be easily corrupted and require extra overhead bits.
- **Unfairness:** The use of small frames means that the channel is only captured for a small amount of time. The adversary however can cheat by quickly responding when needing access while other nodes delay.
- **Exhaustion:** Time division multiplexing can solve the problem of indefinite postponement during collisions. MAC admission control rate limiting is a measure by which the link layer can ignore excessive requests without having to send radio messages.
- **Spoofed, altered or replayed data:** outsider attacks can be prevented in a system by introducing link layer encryption and authentication. This would make Sybil

and selective forwarding attacks trivial. However, these techniques prove useless when the network is confronted with insider attacks. More sophisticated measures need to be applied in such cases, and this is where our research intends to focus.

- Selective forwarding: The solution to selective forwarding attacks is to introduce redundancy to the network in the form of multi-path routing. In such circumstances a message is routed over n paths, and hopefully one message will travel along a path that is disjoint from the selective forwarding node. This is of course given that the number of compromised nodes in the network is less than n .
- The Sybil Attack: An insider cannot be prevented from participating in a network. However, he should only be allowed to do so using identities he has compromised and no more. The solution here is to verify the identities of participating nodes. This can be done by having each node share a unique key with the base station. Two neighbouring nodes then communicate with each other using a shared key to encrypt and verify the link between them. Note that this technique does not stop a compromised node from communicating with the base station or other aggregations points but it definitely limits the number of legitimate nodes the compromised node can communicate with. The base station can help enforce this principle by limiting the number of verified neighbours a node can have and generating an error when this number is exceeded.
- Wormholes and Sinkhole attacks: These are the hardest attacks to defend against because wormholes use channels that are invisible to the network and the advertised routes of sinkholes are extremely hard to verify. Geographic routing protocols however are resistant to these attacks because messages are routed towards the physical location of the base station. False links will be detected by neighbours that figure out that the physical distance of an advertised route exceeds the radio signal range of motes. Another solution would be to provide tight time synchronisation which is often not feasible and requires original protocol design by which to make these attacks useless.
- Hello flood attacks: Such attacks can easily be avoided by verify bi-directionality of a link before taking action based on the information received over that link. The Needham-Schroeder verification protocol [15] does just that. If the base station limits the number of verified neighbours it can prevent this attack all together.
- Acknowledgment Spoofing: The most obvious solution to this problem would be authentication via encryption of all sent packets and also packet headers. The idea is that no node should be able to spoof messages from the base station, yet every node should be able to verify the identity of the base station.
- Flooding: Client puzzles are a way of reducing the severity of flooding attacks by asking all client nodes to demonstrate their commitment to the resources they require. The server distributes a puzzle with each connection request that the client must solve in order to get a connection. The attacker now has to expend more energy to flood the network. The disadvantage is that legitimate nodes now have to expend extra resources to get connected.
- Desynchronisation: The only solution for desynchronisation is to authenticate all packets sent, including control fields. Once again this leads to expenditure of resources for legitimate nodes.

1.7 Intrusion Detection and Classification

In this section, we present background information to some of the techniques that we are considering using in our proposed research. Namely we will be looking at Intrusion Detection Systems (IDS) and the classifiers that make them reality. We will also introduce game theory principles, graph theory and honeypots.

1.7.1 Intrusion Detection Techniques

An Intrusion Detection System generally detects unwanted manipulation to systems [16]. They are applied to monitor a number of points within the system they are protecting. *Passive* IDSs detect potential security breaches and log the information after raising an alarm, whilst *Reactive* systems react to threats either autonomously or at the command of an operator.

The two most common intrusion detection models used in network security today are misuse detection (a.k.a. pattern matching) and anomaly detection .

a) Misuse detection [17] entails identifying and storing signatures of known intrusions and then matching the activities occurring on an information system to these signatures, in order to detect whether the system is undergoing an attack or not. The benefits of this technique are that the signatures are based on well known intrusive activity and hence the attacks detected are well defined. Other benefits include the simplicity of these systems and the ability to detect attacks immediately after installation. In contrast, the major drawbacks of misuse detection systems is their inability to detect unpublished attacks and their reputation of circumventing false negative alarms.

b) The anomaly detection model [17] establishes a profile of the subject's normal activities (norm profile) and then compares activities on the information system to this norm profile. It signals an intrusion when the observed activities, differ largely from those usually undertaken by the user. The major benefit of such techniques is that they can detect attacks that are unpublished; however such systems are complex and resource hungry as they are constantly generating logs and checking audit files.

In our research we will be investigating the powers and shortcomings of both techniques to perform effective intrusion detection for wireless sensor networks.

1.7.2 Artificial Neural Networks

Artificial neural networks (ANNs) are a uniquely powerful tool in multiple class classification, especially when used in such applications where formal analysis would be very difficult or even impossible, such as pattern recognition, nonlinear system identification and control [17]. For the above reasons ANNs and other classifiers have been basis of the traditional IDSs.

What makes ANNs interesting is their ability to learn. Essentially, they are a mathematical model that defines the function [18];

$$f : X \rightarrow Y \tag{1}$$

This means that given a input set X, by applying the neural network function f we acquire a set of outputs Y which classify the input set X. Common neural network functions f are usually a combination of *exponential* or *tanh* functions of the inputs and their associated costs.

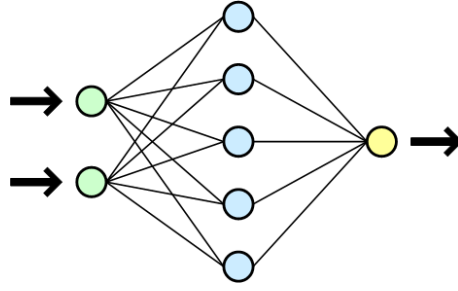


Figure 7: An Artificial Neural Network

Neural networks are characterised as networks because this function is decomposable into a simpler interconnected entities. The type of ANN is defined by the class of such functions that it corresponds to. The crux is that given a class of functions F and a task to solve, learning from a set of observations will let us identify

$$f^* \in F \quad (2)$$

which will solve the task optimally. In order to find the optimal solution however, one must define a cost function such that

$$\forall f \in F : C(f^*) \leq C(f), C : F \rightarrow \mathfrak{R} \quad (3)$$

The cost function is a very important concept in learning as the learning algorithm will search through the solution space to find a function that will produce the smallest possible cost for the task at hand.

There are 3 modes of learning for ANNs:

a) In *supervised learning* we are given sets of data/label pairs and the task is to find the function of that data that matches the given sets.

$$(x, y), x \in X, y \in Y \quad (4)$$

In other words supervised learning infers the mapping between inputs x and outputs y (labels).

b) In *unsupervised learning*, we are given data without the associated labels. The cost function that we are seeking to minimise can be any function is dependant on the input data x and the output function f i.e. the cost function can be deduced from the task we are trying to model and out priori assumptions.

c) In *reinforcement learning*, the data x is not given but rather generated by an agent's interaction with the environment i.e. each point in time t the agent performs an action y_t and the environment generates the observation x_t and an instantaneous cost c_t according to some unknown parameters. The overall goal here is to find a policy for selecting actions that will minimise the long term cost.

1.7.3 K-means nearest neighbours

The K-means nearest neighbour classifier is a simple and popular classifier that uses statistical classification to cluster information into specific groupings, based on a specific distance measures [17].

The main idea of this technique is to define k centroids or means one for each cluster. These centroids are to be associated with a training set initially. The next step has to do with taking each individual point in the training set and associating it with the nearest centroid. This results in each point being assigned to a certain cluster. A graphical representation of this method is shown in *Figure 8*, where there are two centroids (represented by triangles) and the clustering of points around these means is visible with the decision boundary splitting the feature space into two distinct regions.

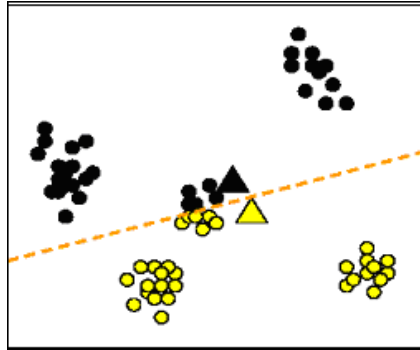


Figure 8: K-means clustering

Common distance measures that are used to determine which cluster a point belongs to are: Euclidean distance, squared Euclidean and Manhattan distance.

1.7.4 Support Vector Machines

Support Vector Machines (SVM) are a set of related supervised learning methods used for classification or regression [19], [20]. These classifiers are fairly novel (1992) and are famous for their quick learning speeds.

When used for classification the SVM algorithm creates a hyperplane that separates the data into two classes with a maximum margin. The points of interest in such models are the points closest to the separating lines also defined as *support vectors*. In order to derive the best possible hyperplane, the idea is to maximize the distance of the support vectors from the hyperplane i.e. increase the separability between the classes [17].

For non-linear classifications the "kernel trick" is applied to maximize margins. In this case, non-linear functions are applied to the data to create a hyperplane in a transformed higher dimensional space i.e. the classifier has a hyperplane in a high-dimension feature space but it may still be non-linear in the original input space. Some common kernels include;

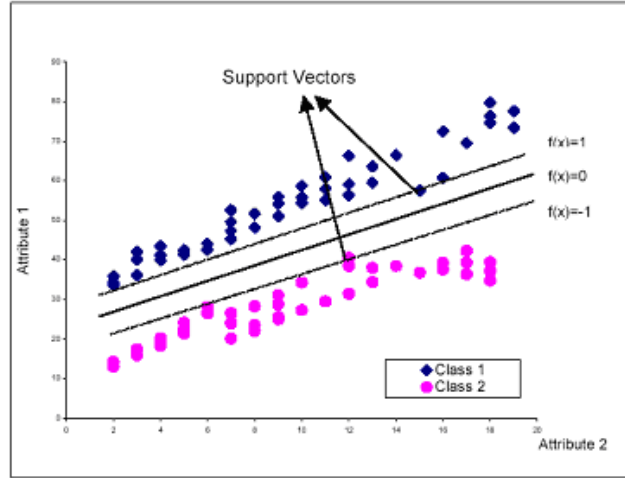


Figure 9: Hyperplane of a Support Vector Machine

- Polynomial (homogeneous):

$$k(x, x') = (x \cdot x')^d \quad (5)$$

- Polynomial (inhomogeneous):

$$k(x, x') = (x \cdot x' + 1)^d \quad (6)$$

- Radial Basis:

$$k(x, x') = \exp(-\|x - x'\|/2\sigma^2) \quad (7)$$

- Sigmoid:

$$k(x, x') = \tanh(\kappa x \cdot x' + c) \quad (8)$$

for $\kappa > 0$ and $c < 0$

1.7.5 Hidden Markov models

Hidden Markov Models (HMMs) are a statistical model where the system being represented is assumed to be a Markov process with unknown parameters [21]. The task at hand is to determine the hidden parameters from the observable parameters. The extracted model can then be used for further analysis, say in pattern recognition applications. Markov Models are usually viewed as Finite State Machines.

In a HMM, each state has a probability distribution over the possible outcomes. Looking at the sequence of outputs generated by the model does not necessarily show the sequence of states. In *Figure 10* the x 's represent the states of the HMM, y 's the are observable outputs, a 's are transition probabilities and the b 's are the output probabilities.

Markov models also evolve as time elapses. Hence t is a important factor in their formulation.

There are 3 problems that can be solved with HMMs;

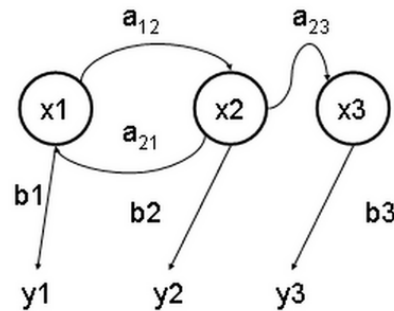


Figure 10: State Transitions of a Hidden Markov Model

- The forward algorithm is used to compute the probability of an output sequence given the model parameters.
- The Viterbi algorithm is used to find the most likely sequence of hidden states to have generated a particular output sequence, given the model parameters.
- Finally, the Baum-Welch algorithm is used to generate the most likely set of state transitions and output probabilities given an output sequence.

1.7.6 Graph Theory

Graph Theory is a very powerful tool when it comes to analysing and understanding large and complex networks. Modelling network traffic and finding the shape of the internet are some of the practical applications of graph theory on networking [22].

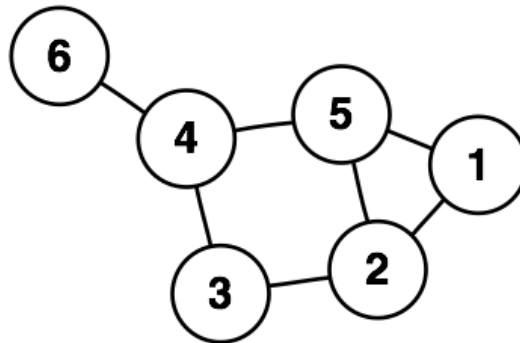


Figure 11: A network Represented as a Graph

In mathematics, graph theory is defined as the study of the properties of graphs. A graph is basically a collection of objects called vertices which are interconnected via links known as edges. For example in a WSN the vertices would represent the sensor nodes and the edges would represent the wireless channel between the nodes. Common graph theory problems include route problems, finding network flow, graph colouring, finding subgraphs etc. Graph Theory will be used in our research to determine optimal placement of intrusion detection agents within a network. See [23].

1.7.7 Game Theory

Game theory is a branch of mathematics where strategic situations are studied, where the players choose different actions in an attempt to maximize their returns [24]. In our situation, game theory can be used to determine the most vulnerable points in a sensor network or even determine the effectiveness of attack recovery techniques.

Games are well defined mathematical objects in game theory. A game consists of a set of players, a set of strategies and payoff for each defined strategy. The game can either be represented in normal form as a matrix or in extensive form as a tree.

There are many types of games:

- *Symmetric* and *asymmetric*: in a symmetric game the payoff depends only the strategies employed and not who is playing the game. Asymmetric games are games in which the strategies are not identical for all players.
- *Zero sum* and *non-zero sum*: In zero sum games the total benefit to all players in the game is zero. Like in poker the gain of one person is the loss of another. In non-zero sum games the net outcome is less or more than zero.
- *Simultaneous* and *Sequential*: A simultaneous game is where the players move at the same time, or if they don't move simultaneously, later players are unaware of the previous players actions. In sequential games the later players have some knowledge about earlier actions.
- *Perfect information* and *imperfect information*: A perfect information game is one where all players know the moves previously made by all other players, hence perfect information games by definition can only be sequential games. The majority of games played however are imperfect information games.
- *Infinitely long games*: pure mathematicians study games that last infinitely long with the winner not known till all moves have been made. Economists and real-game players however play games that are finished in a finite set of moves.

1.7.8 Honeypots

Honeypots are decoy computer resources that are used for the purpose of monitoring and logging the activities of entities that probe, attack or compromise them [?].

Essentially, honeypots are a trap set to detect, deflect or counteract attempts at unauthorised use of information systems. They usually consist of a computer, data or network site that appears to be part of a network but which is really isolated and contains information that would be of value to an attacker. Honeypots are usually valuable surveillance tools and provide early warnings to system administrators about the trends of malicious activity in their networks. Honeypots come in many shapes and sizes i.e. dummy items in a database, preconfigured network sinks, hosts with real operating systems and services and even unused IP address spaces. Honeypots have no production value for legitimate traffic or activities and hence it can be assumed that data they capture is malicious and unauthorised. One practical application is the use of honeypots to thwart spam by masquerading as systems that are usually abused by spammers. We will be looking at including honeypots in our research to help identify the attack signatures that will be used by the IDS.



Figure 12: Winnie-the-Pooh Lured by the Honey

2 Research Aims

Our research is aiming at developing a hybrid model for *fault tolerant* and *energy efficient intrusion detection* for wireless sensor networks. This system will be developed as a security solution for real-life sensor network hardware.

As hinted in the introduction, we will looking at intrusion detection as a form of protection against malicious insider attacks, with the hope that such a system will also provide some protection against outsider attacks.

Our goal is to extend the lifetime of a network under attack for as long as possible. Hence we will be looking at finding an energy efficient and accurate classifier to take on the job of detecting malicious activity on the network. We will also be using game theory strategies in combination with graph theory principles to determine the most effective points in the network to place these intrusion detection agents. Such a hybrid system will also incorporate recovery techniques that will help the system recover and overcome launched attacks.

We expect the final system to be applied successfully to both applications with high volumes of malicious activity and also applications where malicious activity is scarce.

3 Literature Review

Research in the area of Wireless Sensor Networks has increased exponentially since the turn of the millennium. Researchers are focused on addressing the myriad of challenges, that have spawned from the limited resource capabilities of the hardware i.e. memory, processing power, bandwidth and energy deposits. In particular, much research is currently being conducted in the following areas:

- Increasing network lifetime
- Improving reliability of data transfer
- Finding solutions to assist easy deployment and maintenance
- Developing techniques that will enforce secure, private and trustworthy networks

In this literature survey, we attempt to present and evaluate the work that has been done on the subject of Security & Privacy for WSNs.

Presently, there are two schools of thought that are being argued in this area; many researchers insist that WSNs will never become secure enough for commercial

use, unless security and privacy measures are considered during the design phase. Such researchers are primarily interested in developing secure protocols from scratch. Others however, state that intelligent security add ons may be more than sufficient, whilst requiring less development costs.

Work from both sides is presented throughout this literature survey.

3.1 Encryption

Sensor Networks mainly operate in public or uncontrolled areas, over inherently insecure wireless channels. It is therefore trivial for a device to eavesdrop or even inject messages into the network. The traditional solution to this problem has been to espouse techniques such as message authentication codes, symmetric key encryption schemes and public key cryptography. However, since motes are severely constrained, the major challenge here is to implement these encryption primitives in an efficient way without sacrificing their strength.

3.1.1 Shared Keys

One method of protecting any network against outsider attacks is to apply a simple key infrastructure. However, it is known that global keys provide no network resilience and pairwise keys are not a scalable solution. A more intuitive solution is needed here for WSNs.

TinySec [25], was developed as a first attempt to introduce security to the link layer of the TinyOS suite. This was done by incorporating software-based symmetric keying with low overhead requirements. Unfortunately, not all vulnerabilities of TinySec have been addressed i.e. how to avoid insider attacks.

In contrast, Zigbee or the 802.15.4 standard [26] has introduced hardware-based symmetric keying with success. Much work however needs to be done to this standard before all the security measures it applies can actually be considered secure.

Some researchers are investigating the possible use of public cryptography to create secure keys during network deployment and maintenance phases [27]. This concept has opened an uncharted territory for the sensor network cryptographic infrastructure.

Extensive research is also being conducted on topics such as key storage & key distribution [28], key maintenance [29] and shared key pools [30].

3.1.2 Secure Groups

Since sensor nodes are required to group themselves in order to fulfil a particular task, it is necessary that the group members communicate securely between each other, despite the fact that global security may also be in use.

Sadly, secure grouping has not been intensively researched in the past and only a few resource intensive solutions exist. Exceptions are the solutions where more powerful nodes are in charge of protecting the members of static groups [31], [32]. Such solutions would nicely compliment the dominance of cluster based protocols such as LEACH [12], PEGASIS [33] and BCDP [34].

3.2 Data Aggregation

In order to reduce overhead costs and network traffic, sensor nodes aggregate measurements before sending them to the base station. Such data is particularly enticing to an attacker. An adversary with control over an aggregating node, can choose to ignore reports or produce false reports, affecting the credibility of the generated data and hence the network as a whole.

The main aim in this area is to use resilient functions, that will be able to discover and report forged reports through demonstrating the authenticity of the data somehow.

Wagner [35], established a technique in which the aggregator uses Merkle hash trees to create proof of its neighbours' data, which in turn is used to verify the purity of the collected data to the base station. An other approach [36], takes advantage of the network density by using the aggregator's neighbours as witnesses. It is also possible to reduce the amount of traffic heading to the base station by using bloom filters to filter out the false aggregations [37].

Improvements still need to be made in this area, such as minimising the amount of negotiation data generated by interactive algorithms.

3.3 Secure Protocols

The main challenge in this area of research, is to discover new protection techniques that can be applied to existing routing protocols, without forfeiting connectivity, coverage or scalability.

Perrig et al [38] made the first attempt to design a secure protocols for sensor networks. This protocol also known as SPINS: (Security protocols in Sensor Networks) provides data authentication, replay protection, semantic security and low overhead. This work has in turn been used to secure cluster based protocols such as LEACH [39].

Karlof and Wagner [15] have provided an extensive analysis on the routing vulnerabilities of WSNs and possible countermeasures (see *Sections 1.5 - 1.6*). According to their study, common sensor network protocols are vulnerable due to their simplicity and hence security should be built into these protocols during design time. In particular, their study targets TinyOs beaconing, directed diffusion and geographic routing. Although this study is the basis for much of the research to follow, the attacks they focus on are still theoretical and have not been implemented practically on any type of hardware. This research has been furthered by Mun and Shin [40], who suggest countermeasures for routing attacks that establish trust relationships between nodes and authenticate sent packets whilst checking node bi-directionality. Other researchers have focused on developing techniques that target specific attacks such as DoS [2] and the Sybil attack [41].

In contrast, Undercoffer et al [42] moved away from routing information and looked at the application layer in order to detect and correct aberrant node behaviour.

3.4 Intrusion Detection Systems

Intrusion Detection Systems (IDSs) are auditory systems, that are able to query the status of the network by receiving information about internal events. They operate by gathering and analysing audit data in order to detect attacks and apply the correct countermeasures, thus constituting a second line of defense. In contrast to the

techniques presented above IDSs, are able to identify both insider attacks and outsider attacks occurring on a network (see *Section 1.7.1*).

Brutch and Ko [43] have surveyed the challenges in intrusion detection for wireless ad hoc networks and have proposed watchdog, control messages, neighbourhood watch and anomaly detection as possible solutions to dynamic source routing attacks. It would be interesting to see how these techniques would perform on the further resource restricted WSNs.

Since it is impossible for every node to have a full powered IDS agent due to resource limitations, the basic problem in this area is how to distribute the intrusion detection agents and their tasks in the network.

Anjum et al [44] have used graph theory in order to optimally place the intrusion detection modules around the sensor network. Agah et al [45] proved that game theory techniques [23] can be applied as a defense technique which will outperform intrusion detection techniques based on intuitive metrics i.e. traffic loads and Markov decision processes.

Anjum, Subhadrabandhu and Sarkar [46] have focused on signature based intrusion detection techniques and found that this technique generates better results when coupled with proactive routing algorithms rather than reactive ones. Loo et al [47] have focused on using clustering algorithms and anomaly detection to detect aberrant behaviour.

Su et al [32] have researched how to apply intrusion detection techniques in cluster based networks, by making nodes aware of packet forwarding misbehaviour of their neighbours and by collectively monitoring the cluster heads.

In a totally different perspective from the intrusion detection norm that is being established in this field, Doumit and Agrawal [48] experimented with using trends in the aggregated data and letting the sensor network adapt to the norm of the dynamics in its natural environment. In a similar fashion, the literature includes partial solutions that check the integrity of the nodes such as code attestation [49] and health monitoring [50]. Finally, Kreibich and Crowroft [51] have described a system for automating attack signature generation that eliminates the costly procedure of audit data analysis on wired networks. On the same note, Han et al [52] discuss using data mining aided methods to generate Signature Discovery systems. These techniques may be extended to provide efficient misuse detection on WSNs.

Interesting results in terms of energy efficiency and detection accuracy may well be produced by combining some of the above aforementioned techniques into hybrid entities.

The point to remember here, is that intrusion detection solutions based on commercial IDSs will not be effective for pure WSNs. Well at least, this is the governing thought amongst the researchers.

Note :- The above research has produced results based on simulations. Results produced from direct application of the developed techniques to physical hardware is yet to be generated.

3.5 Privacy

Sensor Networks are systems that rely on the collection of information to perform their tasks. Therefore, an additional system requirement is that guidelines regarding fair information practices are built into the networks, in an attempt to protect privacy

rights. To elaborate, content, identity and location privacy of the network need to remain intact for a system to be considered 'private'.

The literature suggests that solutions such as data encryption, access control, the anonymous storing of data and distributed query processing might be the way to go.

Olariu et al [54] take a good stab at privacy issues by defining schemes that maintain the anonymity of the virtual infrastructure of a WSNs. This was coupled by randomising communications, such that the cluster structure and coordinate system remain concealed to outside observers.

This area still remains vastly unexplored. Scenarios need to be explored where privacy is being exploited and solutions need to be devised to solve these issues.

3.6 Other Issues

Due to the immaturity of sensor networks as a networking solution, there is a plethora of security applications that have not yet been fully investigated. Such an example is the use of mobile agents, which are a powerful tool for collaborative processing. It is therefore crucial that a network is able to identify and authenticate these agents and the instructions they deliver to the network, else it would be very easy to inject false information into the nodes or modify collected results. Solutions such as securing code and securing results in constrained environments need to be considered.

3.7 Open Research Areas

Research in the field of WSNs is growing rapidly and achieving tangible results that apply to real life scenarios. However, this field is still at its infancy and there is much room for improvement. Public key cryptography and intrusion detection and reaction are fairly new areas. Secure data aggregation algorithms need to be optimised and secure routing algorithms need to comply with the coverage, connectivity and fault tolerance requirements of the networks. Also privacy of information flow needs to be addressed.

To summarise, research attention needs to be directed to the following areas [4]:

- Tolerating the lack of physical security
- Optimising the security infrastructure in terms of resources (energy and computation)
- Detecting and reacting to DoS attacks
- Raising the issue of social privacy problems
- Management and protection of mobile nodes and base stations.
- Secure administration of multiple base stations with delegations of privileges.

During this PhD, we will be addressing the issues of energy efficient intrusion detection and prevention techniques in collaboration with already established secure routing algorithms (dot points 2& 3). We believe that there are many interesting results to be gained from combining some of the surveyed techniques into an overall effective security solution for WSNs. For more information refer to the proposed research section (*Section 5*).

4 Preliminary Research

In this section we describe the preliminary research to date. The major concern during this stage was to identify an appropriate data source that can be used throughout the entire project. This source is required to replicate both normal and malicious network activity for future analysis and classification.

4.1 Inadequacy of Available Simulation Tools

Simulation tools are very popular in telecommunications as they allow researchers to monitor the behaviour of networks without the cost of developing network prototypes. Simulations also allow researchers to expose the effects of network scalability which is a very important consideration for WSNs.

The two most popular open source network simulators currently used in the area of telecommunication are NS-2 and Omnet++.

NS-2 [55] is a discrete event simulator that provides substantial support for TCP, routing and multicast protocols for wired and wireless networks. Code in NS-2 is developed in a combination of C and a scripting language called TCL.

OMNeT++ [56] is a newer discrete event simulator based on C++. Its primary use is the simulation of communications networks but because of its flexibility and generic architecture it can also be used to simulate IT systems, queueing networks and hardware. Many open source mobility and ad-hoc simulations exist for this framework. OMNeT++ is quickly becoming very popular amongst researchers.

In an attempt to gather preliminary data sets. The following open source simulators were examined.

- *NS-2 Sensor Simulator* [47]: This simulator was developed by students at Melbourne University working on security for WSNs. However, the simulator was found to be flawed with multiple errors and hence data sets were never generated for this program. Also the static nature of NS-2 didn't simplify the procedure of modifying the base protocol functions for further use.
- *LSU SensorSim* [57]: This simulator was developed by Louisiana State University in an attempt to create a simulation platform that will explore both the networking issues and computing aspects of WSNs. This simulator is based on the OMNeT++ framework. However, the distributed software v3.0 contained bugs, that hindered installation. Developers were unable to assist with these problems. Also the underlying structure of SensorSim runs on 802.11 simulations which are not consistent with the standards used by the accessible ambient hardware.
- *EYES WSN Simulation Framework* [58]: This simulator was developed by the University of Twente in Holland and is an attempt to introduce a mobility to sensor network simulations. Based also on OMNeT++ this simulator looks at replicating the behaviour of the μ node of Ambient Systems. However, modifying the code for static network simulations proved messy and inefficient.

In hindsight, the novelty of the security research area became apparent when searching for a suitable tool for the generation of data. The final realisation was that a program would need to be developed that would fit our particular area of research. OMNeT++ is chosen over for the development of this program over NS-2 because of its modular and flexible nature. Steps in this direction are highlighted in the following section.

4.2 Development of a Preliminary Simulator

After spending months investigating available sensor simulator tools and discovering that they are inappropriate for our proposed experiments, we started developing a simple OMNeT++ simulation for the generation of the required data. This simulator is detailed below.

The simulator was based on a template acquired from the WSN Codesign Project on the OMNeT++ website [56]. This template allows for a WSN simulation of 20 static nodes, with the base station periodically broadcasting messages that are flooded throughout the network.

This simulation was modified to as follows;

- A phenomenon was introduced to the simulation which the network is required to detect. This phenomenon is a moving object that randomly progresses through the sensor field. When this phenomenon moves into the vicinity of an active sensor node, the node generates a message that its neighbours are required to forward to the base station. Physical intrusion detection applications, such as tracking tanks and soldiers in the battlefield constitute such network behaviour.
- The flood routing was replaced with geographic routing. A node forwards data packets towards the location of the base station. It will choose the hop that will take the packet closer to the base station. This distance is calculated using manhattan distance measures. To avoid packets getting caught in routing loops, a 25% probability of choosing a random route is introduced. The path a packet follows on its way to the base station are coloured, to allow for visible tracking of messages through the network.
- The size of the network is increased to include 100 sensor nodes that span a larger sensor field than that in the original template. The transmission range of the nodes is decreased to introduce further diversity to the topology.
- The base station was modified to record various parameters of the packets it receives. These parameters are the features of the vectors the classifiers will be using for categorising. Initially, we consider the following features: Source address of the packet, hop count to the base station, the number of packets a particular source generated and the number of packets the base station received from that source node. Total packets in the network and average packet lifetime are also considered.
- A malicious node (hacker) was introduced into the 100 node network. This hacker participates in the network as would a compromised node. When a packet is forwarded to the hacker it is dropped causing a black hole effect.

The developed simulator is far from complete at the moment. All attacks listed in *Section 5.1.2* need to be simulated and multiple hackers with laptop class resources need to be considered. Additional routing protocols such as clustering algorithms also need to be implemented. Lastly energy measures need to be recorded for added realism.

4.3 Preliminary SVM classification

A preliminary classification of the data generated from the above simulator was conducted. In this classification we used a single class SVM which are common to anomaly

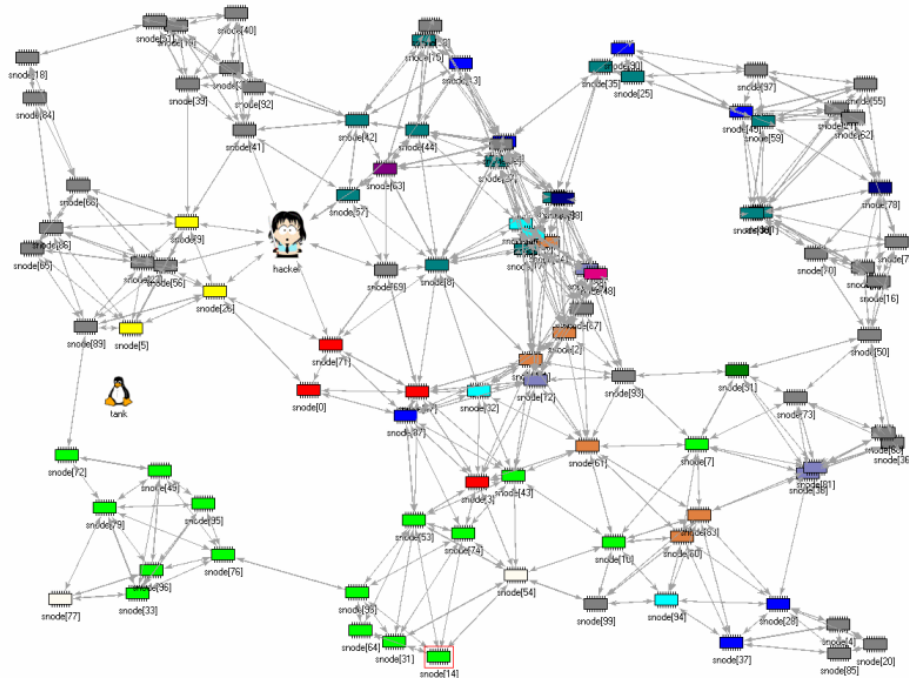


Figure 13: Screenshot from the Preliminary Simulator

detection. Much like a K-means Nearest Neighbour classifier, a single class SVM tries to minimise the area which encompasses all the training points. The training data used was only of normal network activity i.e. single class training data.

Initially the simulator was used to generate a data set of normal traffic data (no black hole activity). This set consisted of 1380 four-dimensional features. The SVM was trained on 1000 of the vectors using an rbf kernel with ν set to 0.7. Out of these points 706 were support vectors. The remaining 380 points of the training set were used to test the SVM in this initial stage. The SVM detected 3/380 anomalies in the normal training set and 377/380 as normal points.

Attack data was then generated by enabling the hacker to drop all packets it received. Out of a total of 1314 generated packets, 101 (7.64%) were dropped by the hacker. The single class SVM was tested on this set and it detected that 8.45% of the vectors in this class are anomalous. In both cases the SVM took approximately 60 seconds to train.

Note: - These results are far from complete. Further tests will need to be conducted with larger data sets before any concrete conclusions can be drawn. However, these initial results aren't at all disappointing for a start.

These results were generated using the program SVM heavy [20].

4.4 Hardware Configuration

Since we are looking at generating realistic data sets, the best way to get such data would be from physical hardware. Results obtained from the hardware can then be compared to those generated by the finalized simulator.

Some initial programming of the Ambient μ nodes has been done in an attempt to

become familiar with the hardware. We have looked at understanding the underlying kernel, programming interrupts, setting timers and displaying results to a computer via a serial connection. Further investigation on enabling the radio chips, setting up network communications, routing and displaying results on the provided LCD screens need to be conducted.

5 Proposed Research

In our literature survey, we identified some gaps in the research, which need to be addressed in an attempt to promote secure and private WSNs. Solutions to WSN security problems are bound to increase their popularity and make them more appealing to commercial applications where security is a concern.

Since the areas of encryption, secure protocols and data aggregation have and are still being intensively researched, it is our suggestion to examine the less common area of auditory systems for WSNs.

In particular, our proposed research will be looking at investigating the following hypotheses:

- Can accurate and energy efficient models be developed for intrusion detection of malicious activity in WSNs?
- Can a fault tolerant model, that allows for recovery of a system under attack, be implemented in WSNs to improve network functionality and lifetime?
- How do such models compare to security solutions that already exist for WSNs.

Figure 14 shows a diagram of the less researched areas of WSN security and highlights the possibility of adapting security solutions used in other types of networks (ad hoc and wired) to fit WSNs. We will be focussing on deepening the knowledge associated with the benefits that Intrusion Detection Systems can bring to WSN security.

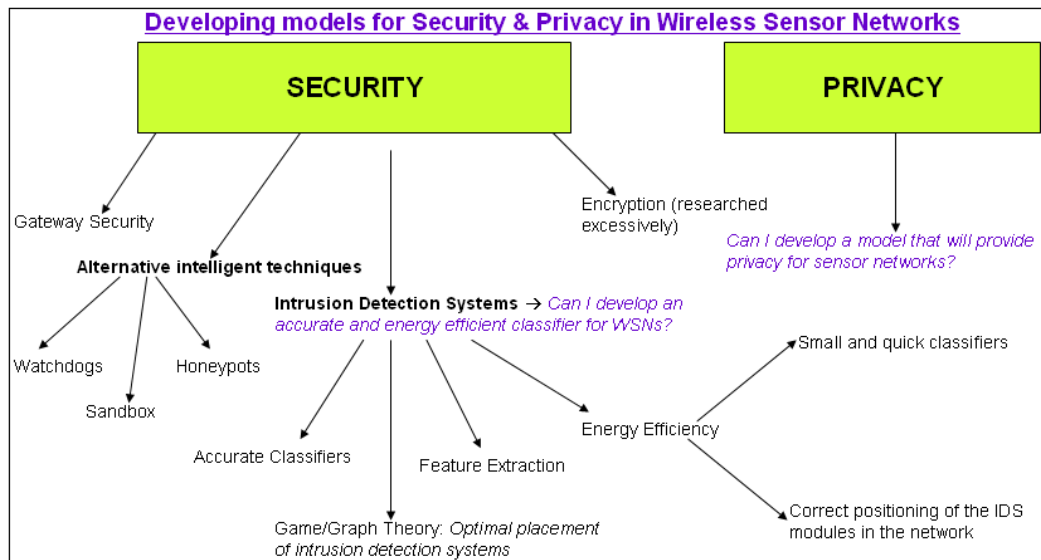


Figure 14: Research Overview

5.1 Research Plan

In this section of the report we provide an explanation of the major tasks that need to be undertaken and the proposed methodologies that will make this feasible. We also depict the resources and timelines required for the completion of the project.

We propose the use physical hardware and simulations to obtain data, test all developed techniques and prove their suitability to real sensor network applications.

5.1.1 The Hybrid Intrusion Detection System

Karlof and Wagner [15], specified the security goals for WSNs that the research community should be aiming for. They state that *ideally*, one would require a security solution that guarantees the integrity, authenticity and availability of all messages even in the presence of attackers, no matter what their power. In the presence of outsider attackers these idealised goals may actually be achievable. However in the presence of insider attackers, especially ones with laptop capabilities, these goals need to be reassessed. So *realistically* we are looking at developing a system that will allow for graceful degradation of the network under attack instead of complete shutdown. This degradation should not propagate through the network faster than a rate proportional to the ratio of compromised nodes to total network nodes.

In an attempt to meet these realistic goals, we propose the creation of a hybrid IDS, which will use optimised intrusion detection and recovery techniques. It is expected that this model will constrain the network as little as possible. The system is classified as a hybrid because it will not only classify attacks but also promote code for recovery from these attacks. Also the IDS modules will be placed optimally in the network for energy efficiency and reinforced defence of vulnerable points. Optimised signature generation techniques to acquire precise high quality signatures will also be incorporated in the design.

Note:- Although IDSs are designed with the intention of discovering misbehaving insiders, we are optimistic that our hybrid system will be able to detect outsider attacks, for which encryption techniques are usually considered as the first line of defense. We will be testing to see whether this expectation holds true.

5.1.2 Attack Replication/Verification

In order to perform misuse detection (*see Section 1.7*), it is necessary to have signatures of the attacks on the network. Given this database of signatures, an IDS can match data packets occurring on the network to those of malicious nature, hence setting alarms in the network. Even anomaly detection techniques require some knowledge of malicious data in order to determine which features of data are more likely to be useful for classification purposes.

Karlof and Wagner [15] and Wood and Stankovic [2] have identified a number of attacks that can be launched on the routing layer of sensor networks (*see Section 1.5*). They agree that WSNs are vulnerable to the following DoS attacks to layer 3 of the protocol stack:

1. Spoofed altered or replayed data
2. Selective forwarding
3. Sinkhole attacks

4. Sybil attack
5. Hello flood attacks
6. Wormholes
7. Black holes
8. Acknowledgement spoofing

The generation of the above knowledge however is not entirely explained in the articles and any suggested solutions thereon are assumed to be based purely on protocol analysis and simulations, bound by a variety of assumptions. Hence, as a side study and a kick-start to this project it would be interesting to rectify which of the above attacks are feasible on WSN hardware and to determine how much damage they actually cause to a network's functionality.

Methodology: In order to replicate DoS attacks on sensor network hardware, the following problems need to be addressed.

a) *Establishing a working network:* Sensor network hardware has recently become available at Melbourne University. Their network consists of 30 Ambient System (Figure 15) sensor nodes that currently don't have any sensor boards attached to them. These nodes haven't yet been configured for any particular application. It is therefore necessary that this hardware be configured to perform a specific application. Initially, we will be considering logging temperature measurements off the microprocessor. Later on additional sensors such as light or vibration sensors can be added to simulate a more realistic applications.

An up and running network will serve as a source of research data and also a test bed for the techniques developed. Data generated from the preliminary simulator will also be used as a secondary source, examining the instances which the hardware cannot cover i.e. scalability of the network.

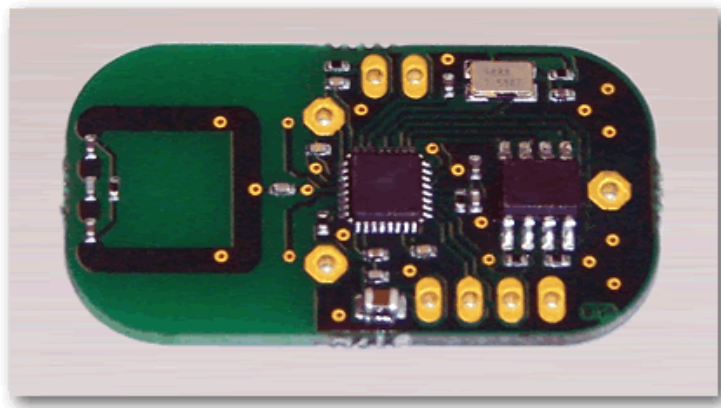


Figure 15: Ambient Systems SmartTag Technology

b) *Data Extraction:* The main challenge of the project is to find an effective method for reliable data extraction from the network. The limited access to physical hardware and the intricacy of this task is possibly the reason why true sensor network information dumps are unavailable on the net, unlike tcp/ip data that the web is swamped with.

To elaborate, we are considering gathering data from the network with one of the two following methods:

- Using regular nodes to intercept all communications in its transmission range. The data they hear can then be relayed back to a computer via a serial cable for logging. In order to do this however, it is necessary to modify the node's code, so that it operates as an eavesdropper and not as a network member. This solution is likely to be the most simple to implement. Multiple loggers may be used in order to collect every packet sent in an operating WSN, that way giving a better overview of the status of the network.
- Using a commercial wireless sniffer. This can be attached to a wireless laptop that can intercept the communications at different points of the network. This method however isn't as favoured as the previous method as much effort will be associated with configuring the wireless receiver of the laptop to the specifications of the wireless chips in the nodes.

Figure 16 below shows how data can be generated for the project. Ideally we would like to extract data from a working WSN, however simulated models and synthetic data can be used to generate data from which further research can be pursued.

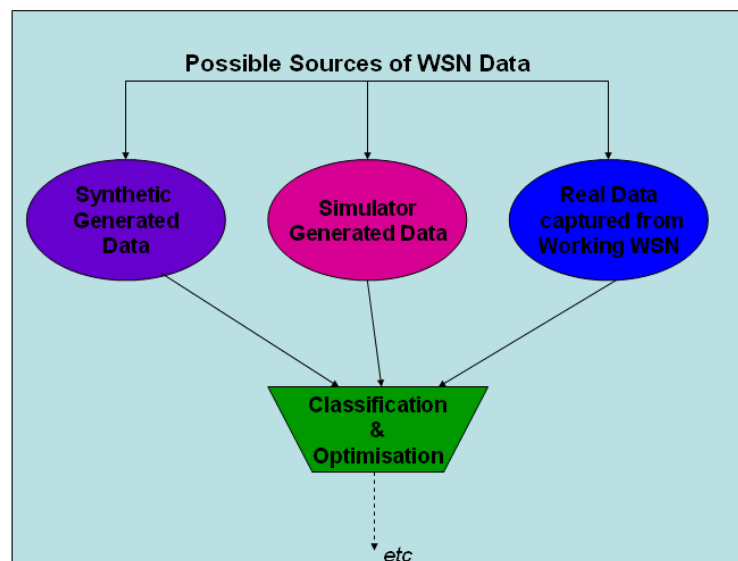


Figure 16: Possible sources of data for network analysis

c) *Attack*: Malicious activity can be replicated by using a laptop with a modified wireless card and running the same protocol stack as the network. Initially though, it may be simpler for the attacker to be a regular sensor node running malicious code. A simple initial attack may be to change the frequency with which the malfunctioning node forwards messages, in an attempt to flood the network. Other more complex attacks will need to be replicated by modifying protocol specific parameters. A contingency plan here would be to generate such attack data synthetically based on known signature analysis techniques common to those used in the internet.

Note :- As outlined in *Section 1.5*, attacks are possible on all layers of the WSN protocol stack. Most researchers accept the fact that the wireless comms are insecure and expect the data link layer (which is specific to wireless radio communications of the chips) to have some sort of security measures already integrated. Since the criticality of the network/routing layer services can be appreciated, we will be focussing on securing this area first. It is expected that in the future research will grow to accommodate for security in all layers of the protocol stack.

5.1.3 Classification

This part of the research, has to do with developing and testing the fault tolerant and energy efficient model that we are proposing. To follow is an outline of the proposed methodology which will make this feasible.

Methodology: a) *Signature Generation:* Both anomaly detection and misuse detection techniques will be trialed to find the most accurate solution for the hybrid IDS.

For the misuse detection part, signatures will be generated automatically using a honeypot like system, as this would be a more efficient solution than wasting time analysing audit data in an attempt to generate attack signatures. It is proposed that the sensor network in its entirety be considered as a honeypot. In particular, a laptop or individual node attacks the network. Specific points in the network relay information back to a computer which will then use string matching techniques, similar to those proposed in Kreibich and Crowcroft [51], to generate the signatures.

b) *Classifiers & optimisation:* We are considering trialing a variation of classifiers on the attack and normal data collected from the network. We will be looking at clustering methods such as K-means nearest neighbours and self-organising maps. Also we will be testing Support Vector Machine methods, Artificial Neural networks and Markov Models.

We will require data sets that contain normal network activity and malicious activity of some sort. These sets will be the training sets for the classifiers. After the classifiers have been trained to identify the difference between the 2 classes accurately we will test them on a data set they have never seen before.

After the preliminary performance results are gathered, attention will be spent on optimising the best classifier such that its resource demands on the network are minimised. This will possibly involve mathematically remodeling the classifier such that the amount of memory and processing power it requires are reduced to a minimum. The goal at this stage is to introduce further improvements to network longevity, especially in the presence of attacks. Online testing and simulation will help to determine the changes if any to network lifetime. Optimisation and testing stages will need to be interleaved recursively to ensure the best results.

c) *Optimal Placement:* In an attempt to further optimise the system, we will be considering how to optimally place the intrusion detection modules around the network. In particular one would need to determine the exact number of agents that are necessary to monitor all possible packets flowing through the network and also which nodes in the network need to be equipped with these agent in

order to do this [44]. This is where graph and game theory techniques will point out the best locations to place the intrusion detection agents in the network.

Nodes with agents will obviously have their energy sources drained faster than those that don't, so schemes for repositioning the agents on nodes with more power need to be considered. In clustering protocols such as LEACH [12], the agents may be placed on all or some of the cluster heads, since the protocol demands that these nodes have higher energy stores than the cluster members. Determining the single weakest point of network to apply the agent to via game theory should also provide interesting outcomes. Mobile code application may need to be considered in this part of the research. The difference optimal placement make on network lifetime will be determined once again via online testing.

5.1.4 Recovery

One thing that many researchers don't consider, is what to do when an attack has been identified by a high accuracy classifier. In this stage of research we will be considering and testing recovery techniques for the replicated routing attacks. One possible method may include purging the malicious node by making legitimate nodes remove the attacker from their routing tables. Another possibility is sending the entire network to sleep for a pre-agreed amount of time, in which way to conserve energy whilst the network is under attack. In an attempt to generate a hybrid security solution such techniques must be included. This will also prove to be one of the most complex parts of the research.

5.1.5 Benchmarking

In an attempt to evaluate the overall performance of the developed system, an in depth comparative study will need to be conducted against other security mechanisms for WSNs. In particular, we will be looking at how our hybrid system competes with encrypting protocols which use key pools and other intrusion detection systems that may have fronted in the area by then.

A trial against the secure protocol TinySec [25] is a must. Other protocols such as S-LEACH and S-MAC are also on the agenda.

Comparative measures that we will be considering include detection accuracy, false positive alarms, energy consumption and most importantly network lifetime under attack over network lifetime without attacks.

5.2 Timetable

The following diagram (*Figure 17*) is an approximate outline of how the major research tasks will be distributed over the next 2 years. It is estimated that the PhD will be completed by late 2008.

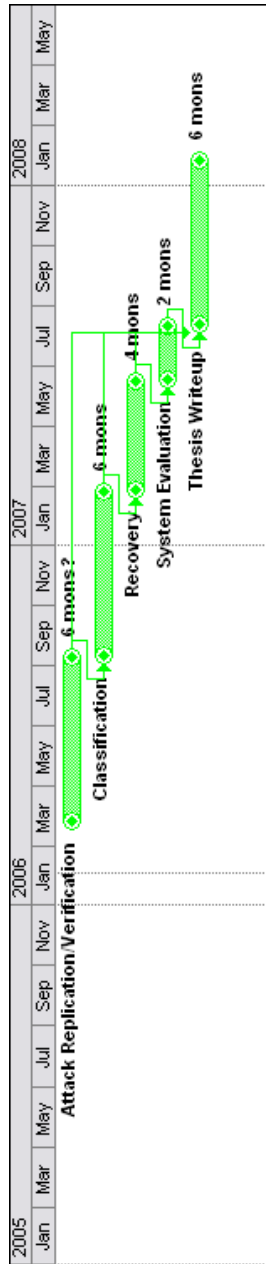


Figure 17: Gantt chart for proposed research

5.3 Resources

The resources associated with the above detailed research are:

- A basic sensor network of 30 odd motes. Initially, 5 motes will be sufficient for the preliminary research. Additional sensors may need to be attached to the motes to enrich the application being tested. In particular, light sensors or vibration sensor are being considered. Results obtained from such a network can be extended to larger networks, given the resources become available.
- An average PC with serial and parallel port capabilities that will act as the

gateway between the WSN and the user. This PC will be used to log data coming off the network and also for analysis of data and the development of classifiers.

- A laptop with wireless network capabilities that will simulate laptop class attacks on the network.
- Software for classifier development including C compilers, SQLdatabase & MATLAB.

All above resources are currently available. Other resources may be required, however at this point in time they are yet to be determined.

6 Conclusions

In this report we introduced the technology of WSNs. In particular we highlighted possible WSN applications, their network architecture, their hardware specifications and their security vulnerabilities. An background to intrusion detection and classifiers was also provided. We also outlined the major DoS threats associated with WSNs.

In the literature survey, we emphasized the work to date conducted by researchers in the area. Much research has been done on the topics of secure routing and wireless encryption. However, simplifying IDSs for such applications is an area yet to be fully investigated.

We propose the development of a fault tolerant and energy efficient intrusion detection model that will increase the lifetime of a network under attack, by incorporating state of the fast classifiers and mathematical strategies. We also intend to transport our system to hardware in an effort to get real performance results. Preliminary research leading towards this system includes developing a realistic WSN simulator, conducting initial attack classifications and getting familiar with the capabilities of the hardware involved.

Publications

S. Kaplantzis and N. Mani, "Classification Techniques for Network Intrusion Detection", in *NCS'06 - Proceedings of the IASTED International Conference on Networks and Communications Systems*, March 2006 (Accepted).

References

- [1] “21 ideas for the 21st century,” *Business Week*, pp. 78–167, Aug. 30 1999.
- [2] A. D. Wood and J. A. Stankovic, “Denial of service in sensor networks,” *Computer*, vol. 35, no. 10, pp. 54–62, 2002.
- [3] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “A survey on sensor networks,” *IEEE Communication Magazine*, Aug. 2002.
- [4] R. Roman, J. Zhou, and J. Lopez, “On the security of wireless sensor networks,” in *International Conference on Computational Science and Its Applications - ICCSA 2005, May 9-12 2005*, vol. 3482 of *Lecture Notes in Computer Science*, (Singapore), pp. 681–690, Springer Verlag, Heidelberg, D-69121, Germany, 2005.
- [5] “Crossbow technology - wireless sensor networks, inertial & gyro systems, smart dust, advanced s,” July 2005. http://www.xbow.com/wireless_home.aspx.
- [6] “Ambient systems - for low cost, low power, wireless mesh networking solutions,” July 2006. <http://www.ambient-systems.net/ambient/index.htm>.
- [7] E. e. a. Sohrabi, “Protocols for self-organization of a wireless sensor network,” pp. 16–27, October 2000.
- [8] A. Woo and D. Culler, “A transmission control scheme for media access in sensor networks,” July.
- [9] E. Shih and et al, “Physical layer driven protocol and algorithm design for energy efficient wireless sensor networks,” July.
- [10] L. Li and J. Halpern, “Minimum- energy module wireless networks revisited,” June 2001.
- [11] W. Heinzelman, J. Kulik, and H. Balakrishnan, “Adaptive protocols for information dissemination in wireless sensor networks,” pp. 174–185, 1999.
- [12] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan, “Energy-efficient communication protocol for wireless microsensor networks,” in *The 33rd Annual Hawaii International Conference on System Sciences (HICSS-33), Jan 4-Jan 7 2000*, Proceedings of the Hawaii International Conference on System Sciences, (Maui, USA), p. 223, IEEE, Los Alamitos, CA, USA, 2000.
- [13] C. Intanagowiat, R. Govindan, and D. Estrin, “Directed diffusion: A scalable and robust communication paradigm for sensor networks,” pp. 56–67, 2000.
- [14] C. Shen, C. Srisatjapornphat, and C. Jaikaeo, “Sensor information networking architecture and applications,” *IEEE Pers. Communication*, pp. 52–59, Aug. 2001.
- [15] C. Karlof and D. Wagner, “Secure routing in wireless sensor networks: Attacks and countermeasures,” *Ad Hoc Networks*, vol. 1, no. 2-3, pp. 293–315, 2003.
- [16] “Intrusion - detection system,” Febraury 2006. http://en.wikipedia.org/wiki/Intrusion_detection_system.
- [17] S. Kaplantzis, “Classification techniques for network intrusion detection,” tech. rep., Monash University, ECSE, October 2004.
- [18] “Neural network,” February 2006. http://en.wikipedia.org/wiki/Neural_network.

- [19] “Support vector machine,” January 2006. http://en.wikipedia.org/wiki/Support_vector_machines.
- [20] A. Shilton, *Design and Training of Support Vector Machines*. PhD thesis, 2006.
- [21] “Hidden markov model,” February 2006. http://en.wikipedia.org/wiki/Hidden_Markov_Models.
- [22] “Graph theory,” February 2006. http://en.wikipedia.org/wiki/Graph_theory.
- [23] M. Kodialam and T. Lakshman, “Detecting network intrusions via sampling: a game theoretic approach,” in *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, vol. 3, pp. 1880–1889 vol.3, 2003.
- [24] “Game theory,” February 2006. http://en.wikipedia.org/wiki/Game_theory.
- [25] C. Karlof, N. Sastry, and D. Wagner, “Tinysec: A link layer security architecture for wireless sensor networks,” in *SenSys’04 - Proceedings of the Second International Conference on Embedded Networked Sensor Systems, Nov 3-5 2004*, pp. 162–175, 2004.
- [26] N. Sastry and D. Wagner, “Security considerations for iee 802.15.4 networks,” in *Proceedings of the 2004 ACM workshop on Wireless security*, pp. 32–42, Philadelphia, PA, USA: ACM Press, 2004.
- [27] D. Malan, M. Welsh, and M. Smith, “A public-key infrastructure for key distribution in tinyos based on elliptic curve cryptography,” in *Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004. 2004 First Annual IEEE Communications Society Conference on*, pp. 71–80, 2004.
- [28] H. Chan, A. Perrig, and D. Song, “Random key predistribution schemes for sensor networks,” in *Security and Privacy, 2003. Proceedings. 2003 Symposium on*, pp. 197–213, 2003.
- [29] W. Du, J. Deng, Y. Han, S. Chen, and P. Varshney, “A key management scheme for wireless sensor networks using deployment knowledge,” in *INFOCOM 2004. Twenty-third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, p. 597, 2004.
- [30] L. Eschenauer and V. D. Gligor, “A key-management scheme for distributed sensor networks,” in *Proceedings of the 9th ACM conference on Computer and communications security*, pp. 41–47, Washington, DC, USA: ACM Press, 2002.
- [31] J. Zachari, “A decentralized approach to secure group membership testing in distributed sensor networks,” In *Proceedings of 2003 Military Communications Conference (MILCOM 2003)*.
- [32] C.-C. Su, K.-M. Chang, Y.-H. Kuo, and M.-F. Horng, “The new intrusion prevention and detection approaches for clustering-based sensor networks,” in *2005 IEEE Wireless Communications and Networking Conference, WCNC 2005: Broadband Wireless for the Masses - Ready for Take-off, Mar 13-17 2005*.
- [33] S. Lindsey and C. Raghavendra, “Pegasis: Power-efficient gathering in sensor information systems,” in *Aerospace Conference Proceedings, 2002. IEEE*, vol. 3, pp. 3–1125–3–1130 vol.3, 2002.

- [34] S. D. Muruganathan, D. C. Ma, R. I. Bhasin, and A. O. Fapojuwo, "A centralized energy-efficient routing protocol for wireless sensor networks," *IEEE Communications Magazine*, vol. 43, no. 3, pp. 8–13, 2005.
- [35] D. Wagner, "Resilient aggregation in sensor networks," *SASN'04 - Proceedings of the 2004 ACM Workshop on Security of Ad Hoc and Sensor Networks*, pp. 78 – 87, 2004. Data aggregation;Sensor networks;Node capture attacks;Multi-party computation;Robust statistics;Average;Mean;Median;.
- [36] W. Du, Y. S. Han, J. Deng, and P. K. Varshney, "A pairwise key pre-distribution scheme for wireless sensor networks," *Proceedings of the ACM Conference on Computer and Communications Security*, pp. 42 – 51, 2003. Wireless sensor networks (WSN);Key pre-distribution;Network resilience;Ultra-small autonomous devices;.
- [37] F. Ye, H. Luo, S. Lu, and L. Zhang, "Statistical en-route filtering of injected false data in sensor networks," *Proceedings - IEEE INFOCOM*, vol. 4, pp. 2446 – 2457, 2004. Sensor networks;En-route filtering mechanism (SEF);Authentication;.
- [38] A. Perrig, J. Stankovic, and D. Wagner, "Security in wireless sensor networks," *Communications of the ACM*, vol. 47, no. 6, pp. 53–57, 2004.
- [39] A. C. Ferreira, M. A. Vilaca, L. B. Oliveira, E. Habib, H. C. Wong, and A. A. Loureiro, "On the security of cluster-based communication protocols for wireless sensor networks," in *Networking - ICN 2005, Apr 17-21 2005*, vol. 3420 of *Lecture Notes in Computer Science*, (Reunion Island, France), pp. 449–458, Springer Verlag, Heidelberg, D-69121, Germany, 2005.
- [40] Y. Mun and C. Shin, "Secure routing in sensor networks: Security problem analysis and countermeasures," in *International Conference on Computational Science and Its Applications - ICCSA 2005, May 9-12 2005*, vol. 3480 of *Lecture Notes in Computer Science*, (Singapore), pp. 459–467, Springer Verlag, Heidelberg, D-69121, Germany, 2005.
- [41] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis and defenses," in *Third International Symposium on Information Processing in Sensor Networks, IPSN 2004, Apr 26-27 2004*, Third International Symposium on Information Processing in Sensor Networks, IPSN 2004, (Berkeley, CA., United States), pp. 259–268, Association for Computing Machinery, New York, United States, 2004.
- [42] S. Avancha, J. Undercoffer, A. Joshi, and J. Pinkston, "Secure sensor networks for perimeter protection," *Computer Networks Wireless Sensor Networks*, vol. 43, no. 4, pp. 421–435, 2003.
- [43] P. Brutch and C. Ko, "Challenges in intrusion detection for wireless ad-hoc networks," in *Applications and the Internet Workshops, 2003. Proceedings. 2003 Symposium on*, pp. 368–373, 2003.
- [44] F. Anjum, D. Subhadrabandhu, and S. Sarkar, "Signature based intrusion detection for wireless ad-hoc networks: A comparative study of various routing protocols," in *2003 IEEE 58th Vehicular Technology Conference, VTC2003-Fall, Oct 6-9 2003*, vol. 58 of *IEEE Vehicular Technology Conference*, (Orlando, FL, United States), pp. 2152–2156, Institute of Electrical and Electronics Engineers Inc., Piscataway, United States, 2004.

- [45] A. Agah, S. K. Das, K. Basu, and M. Asadi, "Intrusion detection in sensor networks: A non-cooperative game approach," in *Proceedings - Third IEEE International Symposium on Network Computing and Applications, NCA 2004, Aug 30-Sep 1 2004*, Proceedings - Third IEEE International Symposium on Network Computing and Applications, NCA 2004, (Cambridge, MA, United States), pp. 343–346, IEEE Computer Society, Los Alamitos, CA 90720-1314, United States, 2004.
- [46] F. Anjum, D. Subhadrabandhu, S. Sarkar, and R. Shetty, "On optimal placement of intrusion detection modules in sensor networks," in *Proceedings - First International Conference on Broadband Networks, BroadNets 2004, Oct 25-29 2004*, Proceedings - First International Conference on Broadband Networks, BroadNets 2004, (San Jose, CA, United States), pp. 690–699, IEEE Computer Society, Los Alamitos, CA 90720-1314, United States, 2004.
- [47] C. E. Loo, M. Y. Ng, C. Leckie, and M. Palaniswami, "Intrusion detection for sensor networks," 2004.
- [48] S. S. Doumit and D. P. Agrawal, "Self-organized criticality and stochastic learning based intrusion detection system for wireless sensor networks," in *MILCOM 2003 - 2003 IEEE Military Communications Conference, Oct 13-16 2003*, vol. 1 of *Proceedings - IEEE Military Communications Conference MILCOM*, (Monterey, CA, United States), pp. 609–614, Institute of Electrical and Electronics Engineers Inc., Piscataway, United States, 2003.
- [49] A. Seshadri, A. Perrig, L. Van Doorn, and P. Khosla, "Swatt: Software-based attestation for embedded devices," in *Proceedings - 2004 IEEE Symposium on Security and Privacy, May 9-12 2004*, vol. 2004 of *Proceedings - IEEE Symposium on Security and Privacy*, (Berkeley, CA, United States), pp. 272–282, IEEE Computer Society, Los Alamitos; Massey University, Palmerston, United States; New Zealand, 2004.
- [50] C.-F. Hsin and M. Liu, "A distributed monitoring mechanism for wireless sensor networks," in *Proceedings of the 2002 ACM Workshop on Wireless Security, Sep 28 2002*, Proceedings of the Workshop on Wireless Security, (Atlanta, GA, United States), pp. 57–66, Association for Computing Machinery, 2002.
- [51] C. Kreibich and J. Crowcroft, "Honeycomb: creating intrusion detection signatures using honeypots," *SIGCOMM Comput. Commun. Rev.*, vol. 34, no. 1, pp. 51–56, 2004.
- [52] H. Han, X. L. Lu, J. Lu, C. Bo, and R. L. Yong, "Data mining aided signature discovery in network-based intrusion detection system," *Operating Systems Review (ACM)*, vol. 36, no. 4, pp. 7–13, 2002.
- [53] O. Kachirski and R. Guha, "Effective intrusion detection using multiple sensors in wireless ad hoc networks," in *System Sciences, 2003. Proceedings of the 36th Annual Hawaii International Conference on*, p. 8 pp., 2003.
- [54] S. Olariu, Q. Xu, M. Eltoweissy, A. Wadaa, and A. Y. Zomaya, "Protecting the communication structure in sensor networks," *International Journal of Distributed Sensor Networks*, vol. 1, pp. 187–203, 2005.
- [55] "The network simulator - ns-2," June 2005. <http://www.isi.edu/nsnam/ns/>.
- [56] "Omnet++ community site," November 2005. <http://www.omnetpp.org/>.

- [57] S. Iyengar, V. Kunchakarra, and A. Suri, "Lsu sensorsimulator - user manual," tech. rep., Department of Computer Science, LSU, January 2005.
- [58] "Eyes wsn simulation framework," April 2005. <http://wwwes.cs.utwente.nl/ewsnsim/>.