Faculty of Science
Math and Computer science Department

PortSaid University

Using Biological Techniques for MANet security based on fuzzy classification

A Thesis submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy in Computer Science

Prepared By

Yaser Maher Abdel-Montaleb wazery
M.SC Information Technology

Supervised by

Prof.Dr. Ibrahim Mohamed Hanafy
Math & Computer science Dept.
Faculty of Science
Port-Said University

Dr. Ahmed Abdel-Khalek Salama
Math & Computer science Dept.
Faculty of Science
Port-Said University

Dr. Mohamed Abdel-Fattah Mohamed
Information systems Dept
Faculty of Computers and Information
Banha University

.

2013

ACKNOWLEDGMENTS

First of all I'd like to thank ALLAH for his non-countable blessings. GOD is the only one deserves the appreciation for doing any good deed in life.

Second to give rights to their owners, I'd gratefully thank my mentors, guiders and supervisors Prof.Dr. Ibrahim Mohamed Hanafy, Dr. Ahmed Abdel-Khalek Salama and Dr. Mohamed Abdel-Fattah Mohamed. For their remarkable efforts to illuminate me through the way for creating this work. I ask ALLAH to give them so many rewards since I can't reward them enough.

Third I'd like to dedicate this work to my father's soul. Dad you are the raw model for people like me, you are the one who isn't here and I want you so much to be here for me as you always did. God bless your soul.

At last I'd like to thank my mother, the endless river of love and giving, my wife who provided a lot of helping and providing a convenient environment for my research. this work is also dedicated for my brother and sister. Finally I'd dedicate this work for my kids the beautiful princess Jodi and the handsome prince Ahmed , I ask ALLAH to see them a better than I am.

Publications

1) I.M.Hanafy, A.A.salama, M.Abdelfattah and Y.M.Wazery, "Security in MANET based on PKI using fuzzyfunction".IOSR Journal of Computer Engineering. India.2012.

2) I.M.HANAFY1, A.A.SALAMA1, M. ABDELFATTAH2 & Y. M. WAZERY. "AIS MODEL FOR BOTNET DETECTION IN MANET USING FUZZY FUNCTION". International Journal of Computer Networking Wireless and Mobile Communications (IJCNWMC) ISSN 2250-1568 Vol. 3, Issue 1, Mar 2013, 95-102

3) I.M.Hanafy, A.A.salama, M.Abdelfattah and Y.M.Wazery, " A PKI based Security Model for MANET using Intuitionistic Fuzzy Function". Submitted

4)) I.M.Hanafy, A.A.salama, M.Abdelfattah and Y.M.Wazery, " A PKI Securing MANETs Using Intutionistic Fuzzy Function as an alternative to negative selection in AIS ". Submitted

Table of Contents

Table of figures

List of tables

Abstract

The rapid and highly increasing need for wireless communications technologies has been a great focus in the recent days. This focus creates new horizons far beyond the internet and its research which takes great amount of the research. Among those horizons the field of Mobile Ad-hoc Networks (MANET), which is experiencing non-preceded growth in its evolution and applications. With a very important characteristic that is "lake of infrastructure" it provides the ability to set up a network temporarily as they needed then disappears whenever they are supposed to vanish.

Security of MANET is a problem that posses more challenges on the research.

Recently, Mobile Ad Hoc Networks (MANETs) is becoming an active area of research. The classical reactive routing protocols for MANETs are: DSR (Dynamic Source Routing) and AODV (Ad-Hoc On-demand Distance Vector Routing) . Similarly research in Bio/Nature routing protocols has resulted in state-of-the-art protocols like AntHocNet, BeeAdHoc and Termite. An important research focus is now on understanding the impact of misbehaving nodes in a MANET environment.

The nature of ad hoc networks poses a great challenge to system security designers due to the following reasons:

1) The wireless network is more susceptible to attacks ranging from passive eavesdropping to active interfering. 2) The lack of an online CA or Trusted Third Party adds the difficulty to deploy security mechanisms. 3) Mobile devices tend to have limited power consumption and computation capabilities which makes it more vulnerable to Denial of Service attacks and incapable to execute computation-heavy algorithms like public key

algorithms 4) In MANETs, there are more probabilities for trusted node being compromised and then being used by adversary to launch attacks on networks, in another word, we need to consider both insider attacks and outsider attacks in mobile ad hoc networks, in which insider attacks are more difficult to deal with 5) Node mobility enforces frequent networking. Reconfiguration which creates more chances for attacks, for example, it is difficult to distinguish between state routing information and faked routing information

.

The security provision in MANETs is a challenge because wireless medium is inherently insecure. All nodes in the transmission range of a node can overhear its transmissions and at the same time initiate spurious transmissions of their own. Therefore, MANETs provide malicious nodes an ideal environment for fabricating and launching different types of routing attacks.

A Botnet is a network of thousands (if not more) of computers under the control of a Botnet owner. Each computer is infected with a malicious program called a bot, which actively communicates with other bots in the Botnet or with several bot controllers to receive commands from the Botnet owner. Botnets usually recruit new vulnerable computers using infection methods from several classes of malware, including self-replicating worms, email viruses, etc. They provide their owners with efficient one-to-many command and control mechanisms, which can be used to order an army of controlled computers (bots) to conduct Distributed Denial-of-Service attacks, email spamming, etc. Botnets have become the most serious threat to internet security.

Abstract

In this research Artificial Immune Systems (AIS) as a powerful paradigm of soft computing is used as a defense system in MANET. It known that AIS is motivated and inspired by the Biological Immune System (BIS). They have been extensively studied to protect a computer system against intrusions by attackers in general and network anomaly detection in particular.

Chapter 1: Thesis Introduction

1.1Problem statement

A Mobile Ad Hoc Network (MANET) is a self-organizing, infrastructureless, multi-hop network. The wireless and distributed nature of MANETs poses a great challenge to system security designers. Ad-hoc wireless networks are collections of autonomous, self-organized, wireless end-user terminals, independent of any fixed infrastructure.

The arbitrary topology of ad-hoc wireless network introduces limitations in communication since it relies on efficient and fare nodes cooperation in order to implement specific routing protocols. These limitations in communication provide a fertile ground for attackers [1][1]`. Ad-hoc wireless networks are vulnerable to packet dropping, packet modification, packet misrouting, selfish node behavior, DOS attack, etc. hence providing security guarantees is rather a difficult challenge.

In Ad hoc networks, each node serves as a routing device, which can forward/receive packets to/from its neighbors. MANETs can operate in both isolation or in coordination with a wired infrastructure. MANETs are increasingly applied in many other applications in areas such as intelligent transportation systems and fault-tolerant mobile sensor grids. Flexibility, self-configurability and easy deployment of mobile ad hoc networks (MANET) are making these networks essential component in future mobile and wireless network architectures.

Although security problems in MANETs have attached much attention in the last few years, most research efforts have been focused on specific security areas, such as establishing trust infrastructure, securing routing protocols, or

intrusion detection and response, none of the previous work proposes security solutions from a system architectural view. The lake of infrastructure poses huge number of challenges in MANET through the perspective of network configuration for example [2]:

1. **Channel vulnerability** – broadcast wireless channels allow message eavesdropping and injection easily.
2. **Node vulnerability** – nodes do not reside in physically protected places, thus easily fall under attack.
3. **Absence of infrastructure** –certification/ authentication authorities are absent.
4. **Dynamically changing network topology** puts security of routing protocols under threat.
5. **Power and computational limitations** prevent the use of complex encryption algorithms

The nature of ad hoc networks introduces great amount of challenges to system security designers due to the following reasons:

- **Firstly**: the wireless network is more susceptible to attacks ranging from passive eavesdropping to active interfering;
- **Secondly**, the lack of an online CA or Trusted Third Party adds the difficulty to deploy security mechanisms;
- **Thirdly**: mobile devices tend to have limited power consumption and computation capabilities which make it more vulnerable to Denial of Service attacks and incapable to execute computation-heavy algorithms like public key algorithms;
- **Fourthly**, in MANETs, there are more probabilities for trusted node being compromised and then being used by adversary to launch attacks on networks, in another word, we need to consider both

insider attacks and outsider attacks in mobile ad hoc networks, in which insider attacks are more difficult to deal with;

- **Finally**, node mobility enforces frequent networking reconfiguration which creates more chances for attacks, for example, it is difficult to distinguish between stale routing information and faked routing information.

Artificial Immune Systems (AIS) is a new paradigm of soft computing which is motivated by the Biological Immune System (BIS). Negative selection algorithm is one of the important techniques in this paradigm that is widely applied to solve two-class (self and non-self) classification problems. Many advances to Negative Selection Algorithms (NSA) occurred over the last decade. This algorithm uses only one class (self) for training resulting in the production of detectors for the complement class (non-self). This paradigm is very useful for anomaly detection problems in which only one class is available for training, such as intrusive network traffic and its detection problem [3].

The use of artificial immune systems in solving security problems is an appealing concept for two reasons. Firstly, the human immune system provides the human body with a high level of protection from invading pathogens in a robust, selforganized and distributed manner. Secondly, current techniques used in computer security cannot cope with the dynamic and increasingly complex nature of computer systems and their security [1]. The strengths of the architecture can benefits many applications which depend on ad hoc technology such as emergency, health-care systems,

groupware, gaming, advertisements, and customer -to- customer applications, and military purposes.

In Botnets A computer waiting for its commander to give it orders is called a bot (or sometimes a zombie). A collection of these bots connected to a network is called a Botnet, but usually we talk of a Botnet when we mean a network of compromised computers which can be controlled by an attacker to e.g. distribute spam mail or start DDoS attacks. This way the original attacker remains anonymous. These computers are usually compromised by malicious software, malware, like viruses or Trojans and wait for their attacker to give them commands what to attack and when. Bots often connect to an IRC network. On this network they join a channel which is operated by the attacker, which gives them their instructions via the channel. Botnets can consist of thousands or millions of hosts and are therefore able to attack in a very distributed and powerful way [4].

An attack is difficult to stop because of its large number of sources. Therefore it is essential to prevent the forming of these networks for example by preventing computers from getting infected or by taking down the central command point the bots are contacting.

Recent malicious attempts are intended to get financial benefits through a large pool of compromised hosts, which are called software robots or simply “bots.” A group of bots, referred to as a Botnet, is remotely controllable by a server and can be used for sending spam mails, stealing personal information, and launching DDoS attacks. Growing popularity of Botnets compels to find proper countermeasures but existing defense mechanisms hardly catch up with the speed of Botnet technologies [5].

1.2 Motivations

Due to the great challenges introduced by MANETs this research was motivated to challenge

Providing adequate security measures for ad hoc networks is a challenging task. In a security concept, typically striving for goals like authenticity, integrity, confidentiality, non-repudiation and availability, authentication of communicating entities is of particular importance as it forms the basis for achieving the other security goals: e.g., encryption is worthless if the communication partners have not verified their identities before. There are five main security services for MANETs: authentication, confidentiality, integrity, non-repudiation, availability.

- **Authentication** means that correct identity is known to communicating partner.
- **Confidentiality** means certain message Information is kept secure from unauthorized party.
- **Integrity** means message is unaltered during the communication.
- **Non-repudiation** means the origin of a message cannot deny having sent the message.
- **Availability** means the normal service provision in face of all kinds of attacks.

Misbehavior nodes disrupt communication, or even make it impossible in some cases. Misbehavior detection systems aim at removing this vulnerability. For this purpose, the use of an Artificial Immune System (AIS) approach is suitable, i.e. an approach inspired by the human immune system (HIS).

The goal is to make an AIS that, analogously to its natural counterpart [6], automatically learns and detects new misbehavior, but becomes tolerant to previously unseen normal behavior. We achieve this goal by adding some new AIS concepts to those that already exist:

(1) the "virtual thymus" which provides a dynamic description of normal behavior in the system;
(2) "clustering" is a decision making method that reduces the false-positive detection probability and minimizes the time until detection;
(3) The process of "danger signal", as a way to obtain feedback from the protected system and use it for correct learning and _nal decisions making;
(4) The use "memory detectors", a standard AIS solution to achieve fast secondary response.

A "Botnet" is a network of computers that are compromised and controlled by an attacker. Each compromised computer is installed with a malicious program called a "bot", which actively communicates with other bots in the Botnet or with several "bot controllers" to receive commands from the Botnet owner, or called "botmaster". Botmasters maintain complete control of their Botnets, and can conduct distributed denial-of-service (DDoS) attacks, email spamming, keylogging, abusing online advertisements, spreading new malware, etc.

Fighting Botnets is often a matter of finding their weak spot: their central point of command, or command-and-control server. This is usually an IRC, Internet Relay Chat, network where all compromised computers connect to, but with the use of P2P technology, this central point of command is nowhere to find: the hosts connect to each other and the attacker only has to

become one of the peers to broadcast his commands over the network. A new detection and fighting method is required to prevent or stop such hazardous networks.

1.3 Thesis Questions

This study seeks to answer these research questions.

- What is meant by security in MANETs?
- How dangerous is Botnets in MANETs?
- Is it possible to implement a secured environment in MANETs?
- How to implement a secured MANETs?
- How to use of PKI over MANETs and is it possible?
- How to distribute security keys efficiently and securely?
- How to protect MANETs against Botnets ?
- Why to use fuzzy and Intuitionistic fuzzy as security manipulators?
- Why the use of PKI over MANETs?

1.4 Thesis Objective

Based on the above research questions, we have many objectives to be achieved:

1- Provide the meaning of security in MANETs as it is a very important and tricky issue in the field of MANETs.

2- Illustrate the state of the art for the status of Botnets and honey-bots and the meaning of zombie networks.

3- Implementing a secured MANET in which data can be sent and received taking into consideration the security challenges in MANETs.
4- Demonstrating the use of PKI and how to overcome the problem of the lack of an online CA or Trusted Third Party.
5- Introducing security models these are capable of distributing security and session keys effectively using PKI.
6- Providing security model for the protection of MANETs against Botnets.
7- Introducing novel ways to make of fuzzy and Intuitionistic fuzzy as security manipulators in the case of key distribution and immunity system as classifiers.
8- Finding reasonable and possible ways to solve the problem of CA in the PKI over MANET
9- Deciding the length of the session key which in turn decides the strength of the encryption technique required in each case along the time of data transmission.
10- Implementing the Artificial Immune System (AIS) as a defence system for the MANET to face dangerous types of threats like Botnets.
11- Creating fuzzy logic function as a classifier in the AIS to face any weaknesses introduced by the negative selection mechanism.
12- Applying Intuitionistic fuzzy logic as an intermediate security mechanism for the problem of determining the length of security keys

1.5 Contributions

The contributions delivered by this study may be divided into 3 directions:

I) Security in MANET based on PKI using fuzzy function: in this point a security scheme is proposed based on Public Key infrastructure for distributing session keys between nodes. The length of those keys is decided using fuzzy logic manipulation for the discrimination between some of the attacks applied over this kind of networks. The proposed algorithm of Security-model is an adaptive fuzzy logic based algorithm that can adapt itself with the dynamic conditions of mobile hosts.

II) AIS model for Botnet Detection in MANET using fuzzy function: the second direction of this research was mainly to handle the Botnets in MANETs so Artificial Immune System (AIS) is used as the defence system to the MANET to face Botnets. Also fuzzy logic plays an important role in this direction of research, in fact it provides a very powerful decision making mechanism by which the AIS could decide whether the incoming/outgoing message is self or non-self

III) A PKI based Security Model for MANET using Intuitionistic Fuzzy Function: the third direction of research focuses on the manipulation of security key management via Intuitionistic fuzzy logic hence proposing a security scheme based on Public Key infrastructure (PKI) for distributing session keys between nodes. The length of those keys is decided using intuitionistic fuzzy logic

manipulation. The proposed algorithm of Security-model is an adaptive intuitionistic fzzy logic based algorithm that can adapt itself according to the dynamic conditions of mobile hosts.

The rest of this thesis is organized as follow; chapter 2 provides survey on MANETs, while chapter 3 provides over view to the field of MANETs security, in chapter 4 The proposed Key Distribution Model for Botnets Prevention in MANETs is provided, the AIS Model for Botnets Manipulation in MANETs Using Fuzzy Function is illustrated in chapter 5, Chapter 6: Securing MANETs Using Intutionistic Fuzzy Function as an alternative to negative selection in AIS is well clarified in chapter 6, finally the experimental results and conclusions are provided in chapter 7.

Chapter 2: Mobile Adhoc Networks Literature review

This chapter provides brief overview to MANETs and the history of this special type of networks. In order to understand this type of networks the sections of this chapter handles the application of MANETs, the routing and its protocols and its architecture

2.1 What is a MANET?

A Mobile Ad-hoc Networks (MANETs) is a collection of nodes that are self configuring (network can be run solely by the operation of the end-users). Nodes communicate with each other by forming a multi-hop radio network and maintaining connectivity in a decentralized manner. Each node in MANETs plays both the roles of routers and terminals. Such devices can communicate with another device that is immediately within their radio range or one that is outside their radio range not relying on access point [5].

Opposed to the infrastructure wireless networks where each user directly communicates with an access point or base station, a mobile ad hoc network, or MANET is a kind of wireless ad hoc network. It is a self configuring network of mobile routers connected by wireless links with no access point. Every mobile device in a network is autonomous. The mobile devices are free to move haphazardly and organize themselves arbitrarily.

In other words, MANETs do not rely on any fixed infrastructure (i.e. the mobile ad hoc network is infrastructure less wireless network. The Communication in MANET is take place by using multi-hop paths.

Nodes in the MANET share the wireless medium and the topology of the network changes erratically and dynamically. In MANET, breaking of communication link is very frequent, as nodes are free to move to anywhere.

The density of nodes and the number of nodes are depends on the applications in which we are using MANET [3].

MANET has given rise to many applications like Tactical networks, Wireless Sensor Network, Data Networks, Device Networks, etc. With many applications there are still some design issues and challenges to overcome.

MANETs are widely used in military and other scientific areas. With nodes which can move arbitrarily and connect to any nodes at will [6], it is impossible for Ad hoc network to own a fixed infrastructure. It also has a certain number of characteristics which make the security difficult.

MANETs are self-organized, temporal networks which consist of a set of wireless nodes. The nodes can move in an arbitrary manner and work as its own opinions. They may join or leave the network with no restrictions. Therefore, MANETs' topologies are dynamic and costly to maintain. Furthermore, wireless channels make the routing and message transmission much more challenging [4]. Nodes of these networks can function as routers that discover and maintain routes to other nodes as well as end-users. They will rely other nodes to relay the messages, which are exposed in an open dangerous situation for any intermediate node to be capable of destroying the integrity or choose as their like to deal with the messages. Last but not least, nodes in MANETs have only limited resource, i.e. Battery power, bandwidth and cpu power. They are usually embedded systems which are produced for certain fixed tasks [4].

MANET is self-organizing, self-discipline and self-adaptive. The main characteristics of mobile ad hoc network are:

- Infrastructure-less: (Dynamic topology) since nodes in the network can move arbitrarily, the topology of the network also changes.

- Bandwidth Limitations: The bandwidth of the link is constrained and the capacity of the network is also variable tremendously. Because of the dynamic topology, the output of each relay node will vary with the time and then the link capacity will change with the link change.
- Power limitations: it is a serious factor. Because of the mobility characteristic of the network, devices use battery as their power supply. As a result, the advanced power conservation techniques are very necessary in designing a system.
- Security limitations: The security is limited in physical aspect. The mobile network is easier to be attacked than the fixed network. Overcoming the weakness in security and the new security trouble in wireless network is on demand Figure 2.1 shows the general form of cellular networks vs. MANETs.

A side effect of the flexibility is the ease with which a node can join or leave a MANET. Lack of any fixed physical and, sometimes, administrative infrastructure in these networks makes the task of securing these networks extremely challenging [7].

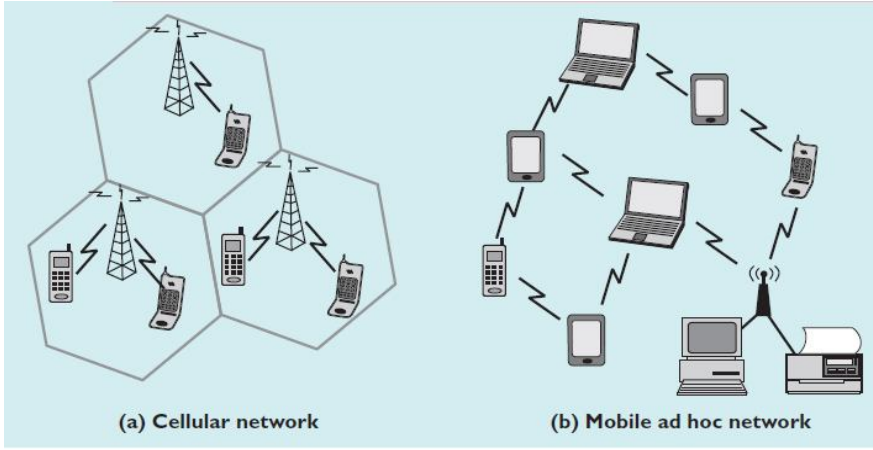


Figure 2.1: Cellular network (a) VS. (b) MANETs

2.2 History of MANETs

The life cycle and evolution of MANET can be characterized into first, second and third generation. Present MANET are considered the third generation [4]. The first generation of ad hoc network can be traced back to 1970's. In 1970's, these are called Packet Radio Network (PRNET) [4]. The Defense Advanced Research Project Agency (DARPA) initiated research of using packet- switched radio communication to provide reliable communication between computers and urbanized PRNET. Basically PRNET uses the combination of Areal Location of Hazardous Atmospheres (ALOHA) and Carrier Sense Multiple Access (CSMA) for multiple access and distance vector routing [5].

The PRNET is then evolved into the Survivable Adaptive Radio Network (SURAN) in the early 1980's. SURAN provides some benefits by improving the radio performance (making them smaller, cheaper and power thrifty). This SURAN also provides resilience to electronic attacks.

Around the same time, United State Department of Defense (DOD) continued funding for programs such Globe Mobile Information System (GloMo) and Near Term Digital Radio (NTDR). GloMo make use of CSMA/CA and TDMA molds, and provides self-organizing and self-healing network (i.e. ATM over wireless, Satellite Communication Network). The NTDR make use of clustering and link state routing and organized an ad hoc network. NTDR is worn by US Army. This is the only "real" ad hoc network in use. By the growing interest in the ad hoc networks, a various other great developments takes place in 1990's [8].

The functioning group of MANET is born in Internet Engineering Task Force (IETF) who worked to standardized routing protocols for

MANET and gives rise to the development of various mobile devices like PDA's , palmtops, notebooks, etc . Meanwhile the Development of Standard IEEE 802.11 (i.e. WLAN's) benefited the ad hoc network. Some other standards are also developed that provide benefits to the MANET like Bluetooth and HIPERLAN.

2.3 Applications of MANETs

2.3.1 Pure general purpose MANET

The mostly discussed application scenario for pure general-purpose MANET is Battlefield or disaster-recovery networks. However, these kinds of networks have not yet achieved the envisaged impact in terms of real world implementation and industrial deployment [9].

Limits of pure general-purpose MANET research

a) **USERS' PERSPECTIVE**

Generally, MANET is justified by the possibility of building a network where no infrastructure exists, or to have a "free" network where users can communicate without cost, provided that the node density is sufficient. However, reports about MANET perception from the users' perspective are missing. The users' evaluation indicates the following major problems in pure general purpose MANET:

• Users' motivations for using large-scale MANET are not clear.

• Application scenarios able to attract user interest are missing.

• There is a lack of effective MANET implementations that can be used by non-expert users.

• Mesh network is a more pragmatic approach to build multihop MANETs.

b) TECHNICAL PERSPECTIVE

Although MANET research has been going on for some time, there are relatively few experiences with real ad hoc network[10]s. The lack of accuracy in most MANET simulation studies in one or more of the previous points drastically reduces the credibility of MANET research. Here are the most common issues in MANET simulation that may result in the lack of realism in simulation studies.

- Simulation Modeling
- Simulation Model Solution
- Analysis of the Simulation Output

2.3 .2 Mesh networks

Mesh networks are built upon a mix of fixed and mobile nodes interconnected via wireless links to form a multihop ad hoc network. Unlike pure MANETs, a mesh network introduces a hierarchy in the network architecture by adding dedicated nodes (called mesh routers) that communicate wirelessly to construct a wireless backbone.

MIT Roofnet provides a city such as Boston, with broadband access with an 802.11b-based wireless network backbone infrastructure [8].

Mesh networks can useful in the following scenarios:

a) Public Internet access.

The wireless mesh networks are the ideal solution to provide both indoor and outdoor broadband wireless connectivity in urban,

suburban, and rural environments without the need for extremely costly wired network infrastructure.

Metroscale broadband city network in the city of Cerritos (California)This network is built up with Troposbased mesh technology and covers a city area as large as eight square miles using more than 130 outdoor access points, less than 20 percent of them directly connected to a wired backhaul network.

This significant reduction of network installation costs ensures rapid deployment of a metropolitan broadband network that is cost effective even with a limited potential subscriber base, as found in rural or scarcely populated urban areas.

b) Intelligent transportation systems

Wireless mesh could be the flexible solution to implement the information delivery system required to control transportation services [11]**.**

Portsmouth Real-Time Travel Information System (PORTAL): aimed at providing real-time travel information to bus passengers in the city of Portsmouth. This system is realized by equipping more than 300 buses with mesh technology provided by MeshNetworks Inc. The wireless mesh network allows anybody to display, at more than 40 locations throughout the city, real-time information on transportation services, such as where his/her bus is, its ultimate destination, and when it is scheduled to arrive. The same system is also expected to be used to address and alleviate transportation congestion problems, control pollution, and improve transportation safety and security.

c) Public Safety

Wireless mesh networks appear to be the natural solution to address the needs of law enforcement agencies and city governments. Currently, several mesh networks are operating to provide public safety applications [12].

The San Matteo Police Department in the San Francisco Bay Area has equipped all its patrol cars with laptops, and motorcycle and bicycle patrols with PDAs, employing standard 802.11b/g wireless cards for communications. The outdoor wireless network is built using mesh networking technology provided by *Tropos* Networks. More than 30 *Tropos* Wi-Fi access points were installed throughout downtown to provide ubiquitous coverage to the zone. *Tropos* proprietary software components are installed over the access points, providing self-discovery and self-configuring functionalities, communications privacy, and centralized network management and control.

d) Mesh community

The Champaign-Urbana Community Wireless Network (CUWiN) implementing a wireless network in the downtown area of Urbana. This is creating a community of users, who install their own nodes and participate in the mesh network that is further supported by other backbone-like nodes [13].

Microsoft research, Intel, Motorola, CISCO have decided to enter the wireless mesh networking.

2.3.3 Vehicular ad hoc networks

VANETs use ad hoc communications for performing efficient driver assistance and car safety. The communications include data from the roadside and from other cars[8]. VANET research aims to supply drivers with information regarding obstacles on the road and emergency events, mainly due to line-of-sight limitations and large processing delays. VANET can be used to communicate premonitions, notification of emergencies, and warnings about traffic conditions.

It can be used for distributing information about road conditions and maintenance, weather forecasts, or other relevant data distribution requirements between vehicles.

VANET enable the use of advanced driver assistance systems (ADAS) and vehicular-to -vehicular (V2V) communications, also called inter-vehicular communications (IVC), as well as communication with roadside infrastructure. VANET have an advantage compared to traditional MANET. They rarely have constraints related to the capacities of the devices.

2.3.4 Applications and Possible scenarios/services[14]:

i) Tactical networks

- Military communication and operations
- Automated battlefields

ii) Emergency services • Search and rescue operations

- Disaster recovery
- Replacement of fixed infrastructure in case of environmental disasters
- Policing and fire fighting

- Supporting doctors and nurses in hospitals

iii) Commercial and civilian

- E-commerce: electronic payments anytime and anywhere environments
- Business: dynamic database access, mobile offices
- Vehicular services: road or accident guidance, transmission of road and weather conditions, taxi cab network, inter-vehicle networks
- Sports stadiums, trade fairs, shopping malls
- Networks of visitors at airports

iv) Home and enterprise

- Home/office wireless networking
- Conferences, meeting rooms
- Personal area networks (PAN), Personal networks (PN)
- Networks at construction sites

v) Education

- Universities and campus settings
- Virtual classrooms
- Ad hoc communications during meetings or lectures

vi) Entertainment

- Multi-user games
- Wireless P2P networking
- Outdoor Internet access
- Robotic pets
- Theme parks

vii) Sensor networks

- Home applications: smart sensors and actuators embedded in consumer electronics

- Body area networks (BAN)
- Data tracking of environmental conditions, animal movements, chemical/biological detection

viii) Context aware services

- Follow-on services: call-forwarding, mobile workspace
- Information services: location specific services, time dependent services
- Infotainment: touristic information

x) Coverage extension

- Extending cellular network access
- Linking up with the Internet, intranets, etc.

2.4 Routing Protocols in MANETs

Existing routing protocols can be classified into mainly two types- proactive routing protocols and reactive routing protocols [7]. Proactive routing protocols such as Destination-Sequenced Distance-Vector Routing (DSDV)[5] maintain routing information all the time and always update the routes by broadcasting update messages. Due to the information exchange overhead, especially in volatile environment, proactive routing protocols are not suitable for ad hoc networks [9]. However, reactive routing is started only if there is a demand to reach another node. Currently, there are two widely used reactive protocols- Ad-hoc On-Demand Distance Vector Routing (AODV) and Dynamic Source Routing (DSR) which will be discussed later. But they all suffer from the high route acquisition latencies [12]. That is, messages have to wait until a route to destination has been discovered.

Normally, reactive routing protocols include two processes- route discovery and route maintenance.

2.4.1 Dynamic Source Routing

DSR is a source routing in which the source node starts and take charge of computing the routes [9].

At the time when a node S wants to send messages to node T, it firstly broadcasts a route request (RREQ) which contains the destination and source nodes' identities. Each intermediate node that receives RREQ will add its identity and rebroadcast it until RREQ reaches a node n who knows a route to T or the node T. Then a reply (RREP) will be generated and sent back along the reverse path until S receives RREP. When S sends data packets, it adds the path to the packets' headers and starts a stateless forwarding [15].

During route maintenance, S detects the link failures along the path. If it happens, it repairs the broken links. Otherwise, when the source route is completely broken, S will restart a new discovery.

2.4.2 Ad-hoc On-demand Distance-Vector

It is similar to DSR when RREQ is broadcast over the network. When either a node knowing a route to T or T itself receives RREQ, it will send back RREP. The nodes receiving RREP add forward path entries of the destination T in their route tables.

There are many differences between DSR and AODV. Firstly, destination T in DSR will reply to all RREQ received while T in AODV just responds to the first received RREQ.

Secondly, every node along the source path in DSR will learn routes to any node on the path. But in AODV, intermediate nodes just know how to get the destination.

2.5 MANET architecture

While capturing important characteristics, this description does not make explicit how MANETs map into the Internet architecture – and does therefore not allow evaluation of existing IP protocols and their applicability on MANETs. Similarly, the lack of a clear architectural description within the context of the Internet has impeded the evaluation of the applicability of MANETs within the Internet. This fact became explicit during the chartering of the IETF AUTOCONF working group: in simple terms, the goal of the AUTOCONF working group is to provide automatic address configuration for MANET nodes. Most researchers and engineers familiar with MANETs shared the understanding that existing autoconfiguration approaches did not apply.

Describing why and how was, absent a clear and agreed upon architectural model of MANETs, difficult – as was communication to experts outside the MANET community [16].

The issue arose again within the context of routing and route optimization within nested NEMO networks, where a clear architectural description of MANETs lead to a poor general understanding of how MANETs might be a candidate technology.

The purpose of this section is to document the MANET architecture within the general Internet and IP architecture.

2.5.1 Classic IP Link and Network Model

Nework protocols and applications are designed with specific assumptions of the nature of an IP link.

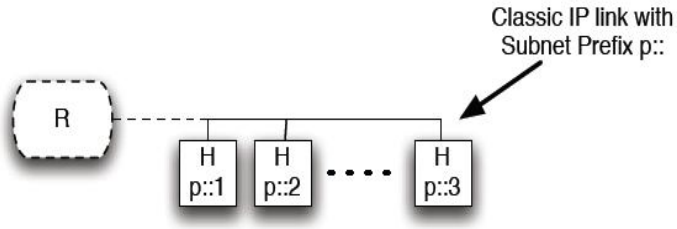


Figure 2.2: Classic IP Link Model: hosts (H) connected to the same link have assigned IP addresses from a common prefix, possibly assigned by a router (R).

Considering figure 2.2, these assumptions can be summarized as follows:

- all hosts (H) with network interfaces configured with addresses from within the same prefix p::, and with the same prefix p:: assigned to the interfaces, can communicate directly with one another – *i.e.*:
  - IP datagrams are not forwarded at the network layer when communicating between interfaces which are configured with addresses from within the same prefix; hence
  - TTL/hop-limit in IP datagrams are not decremented when communicating between interfaces which are configured with addresses from within the same prefix, and;
  - IP datagrams with a TTL/hop-limit of 1 are (modulo data loss) delivered to all interfaces within the same subnet.
- Link-local multicasts and broadcasts are received by all interfaces configured with addresses from within the same prefix without forwarding [17].

An even shorter summary of the *"classic IP link model"* is to say that"an IP link looks like an Ethernet".

It follows from the above that the notion of "IP link" is tied with the notion of an "IP Subnet" (IPv4) or a prefix (IPv6), in that all interfaces which are configured with the same subnet address or prefix are considered to be on the same IP link and thus that for communication between nodes on the same subnet, no forwarding is required and no decrement of TTL/hop-limit is performed.

Interfaces within the same prefix or, for IPv4, within the same subnet, are within the classic IP link model assumed to also be attached to the same classic IP link as described above. For completeness, it should be mentioned that the inverse is not necessarily true: in some network configurations, interfaces connected to the same classic IP link may be configured within different prefixes or subnets [18].

2.5.2 MANET Interface Characteristics

MANET nodes are equipped with MANET interfaces, which have different characteristics than the interfaces described for the classic IP Link and Network Model. These characteristics are briefly summarized in this section, with the purpose of exemplifying the difference with"Ethernet-like" interfaces. A MANET version of Figure 2.2 looks as in Figure 2.3.

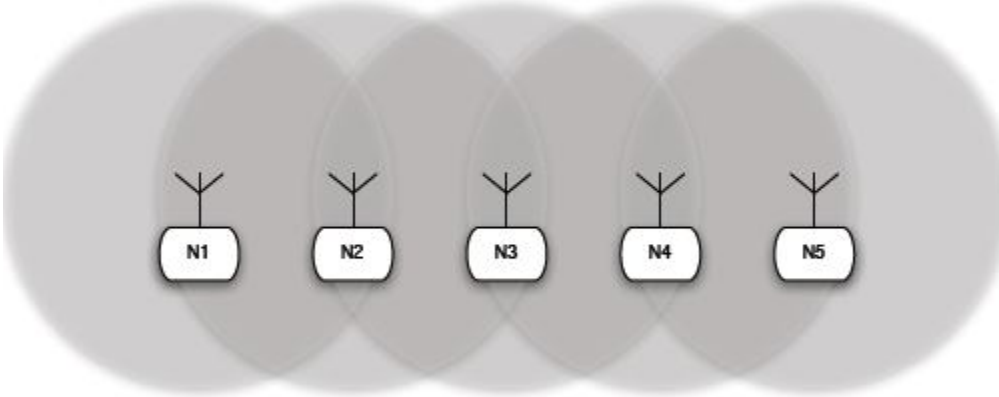


Figure 2.3: MANET: nodes (N) with MANET interfaces. The light grey area indicates the coverage area of each MANET interface.

a) Semi Broadcast Interfaces

Each MANET interface is a broadcast interface, typically, but not necessarily, wireless, which is able to establish a direct L2 connection with only those nodes which are within its coverage area. In figure 2.3, this coverage area is approximated by a simple disc of fixed radius [19]:

– In the real world, both the shape and size of the coverage area is variable as a function of the interface, interference from the environment etc. Referring to figure 2.3 if, for example, if N3 transmits, then this transmission may be received by N2 and N4, but not by N1 and N5. This implies that, *e.g.*, N3 and N4 – despite being neighbors and on the same ”link”

– do not share the same view of which other nodes are neighbours and on the same ”link”: N3 considers that it is on

the same ”link” as N2 and N4, whereas N4 considers itself to be on the same ”link” as N3 and N5.

This sometimes leads to describing MANET interfaces as ”semi-broadcast interfaces”, with non-transitive neighbor relationships: neighboring nodes may experience distinctly different neighborhoods.

b) Shared Bandwidth

Depending on the radio technology used, MANET interfaces may interfere with each other [20] this is for example the case with the commonly used IEEE 802.11 interfaces. In Figure 2.4, if N3 transmits over its MANET interface, then this may cause N2 and N4 to be unable to transmit concurrently over their respective MANET interfaces. The direct consequence hereof is, that available bandwidth is shared among the MANET interfaces within the same coverage area.

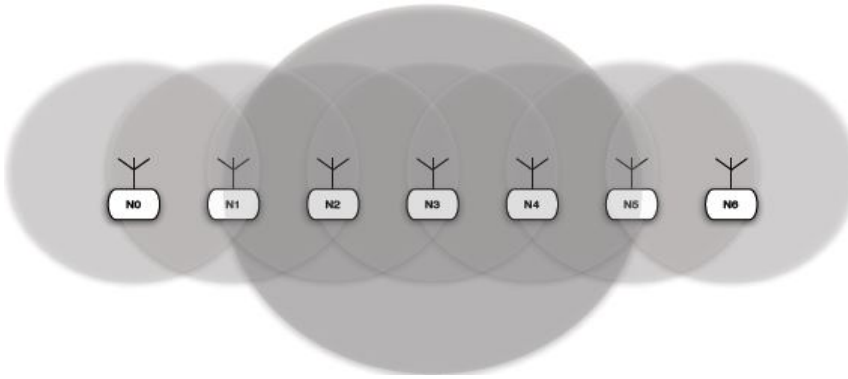


Figure 2.4: MANET: nodes (N) with MANET interfaces. The light grey area indicates the coverage area of each MANET interface. The dark grey circle indicates the interference area of the MANET interface of N3.

A further consideration is, that a wireless interface has an ”interference area” which may be greater than its coverage area, *i.e.* a transmission by N3 in figure 2.3 will, as indicated above, be correctly received by the interfaces N2 and N4. At the same time, however, this transmission may be propagating to interfaces of N1 and N5 where, while the transmission can not be correctly decoded, it can be detected, and cause interference with other transmissions which could otherwise be correctly received over the MANET interfaces of N1 and N5 (such as transmissions from N0 and N6).

c) Hidden Terminals

A property of MANETs which is commonly brought forward is the ”hidden terminal problem”: if N3 through some protocol agrees with its neighbors (N2 and N4) that it will, for the moment, have exclusive access to the wireless media via its MANET interface, then N3 may go ahead and make a transmission. However, if at the same time N1 also transmits over its MANET interface, then the transmissions of the MANET interfaces of N1 and N3 may appear concurrently at the MANET interface of N2 – potentially interfering and causing N2 to receive neither of the transmissions. Denoted a ”collision”, the possibility and probability of this occurring depends on the L2 (data link layer) mechanisms in place suffice to observe that the such collisions can and do occur when using some common wireless interfaces such as IEEE 802.11 [21].

The term ”hidden terminal” originates from the fact that while the node wishing exclusive access to the wireless media may negotiate this with its direct neighbors (in our case N2 and N4), whereas nodes out of direct radio range (in our case N1 and N5) are ”hidden”.

d) Asymmetric Connectivity

Considering Figure 2.2, an axiomatic assumption is that neighbor relationships are symmetric: if communication from one interface to another interface is possible in one hop, then communication in the inverse direction is also possible – in other words, connectivity between neighbor interfaces is symmetric [22]. Considering the small MANET in Figure 2.5: for some reason (powerful transmitter, large antenna, ...) the MANET interface of N1 has a large enough coverage area that its transmissions can be received by the MANET interface N2. The MANET interface of N2, on the other hand, has a much smaller coverage radius, such that transmissions from the MANET interface of N2 do not arrive at the MANET interface of N1. Thus an asymmetric – or more precisely, an unidirectional – connectivity between the MANET interface of N1 and the MANET interface of N2 exists: N2 sees N1 as a neighbour (since the MANET interface N2 can receive transmissions from the MANET interface of N1), whereas N1 does not see N2 as a neighbor (since the MANET interface of N1 can not receive transmissions from the MANET interface of N2). Thus, MANET neighbour relationships are non-reflective. N1 N2

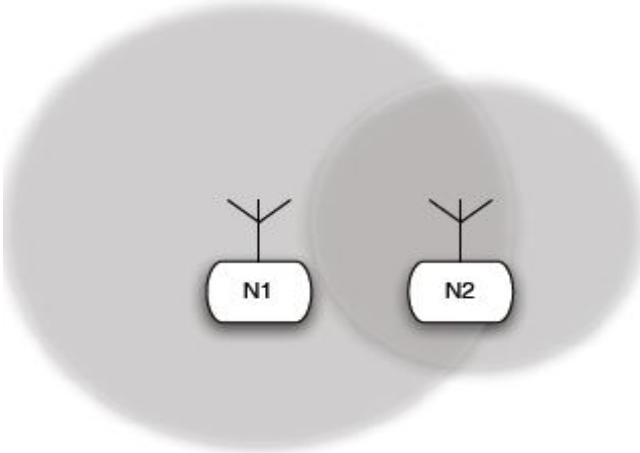


Figure 2.5: MANET: neighbour asymmetry

e) Neighborhood & Network Membership

Returning to the initial description of a MANET in the introduction, MANET interface form *"a dynamic, arbitrary graph"* among themselves. This indicates that the neighborhood of a MANET interface is dynamic and varies over time – either due to node mobility or due to environmental factors which impact the area of coverage of a MANET interface [23]. On a larger scale even the MANET membership may be time varying, with MANET interfaces appearing and disappearing over time, and for the same reasons.

2.6 Common MANET Misperception

Considering the classic IP link model, a common misperception is that "a MANET should emulate an Ethernet at L3", and that the nodes in a MANET are "hosts" [24]. This has lead to MANET nodes being perceived and configured as indicated in figure 6 as hosts in an Ethernet: the MANET

interface is assigned an IP address and a subnet prefix p:: – a prefix which is shared among all the nodes in the MANET as indicated in Figure 2.6.

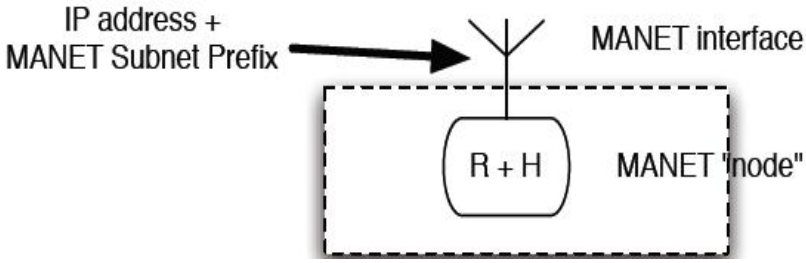


Figure 2.6: Common Misperception of MANET Nodes: viewing MANET nodes as regular hosts in a subnet, with an IP address and a subnet prefix assigned to their MANET interface.

Configuring a MANET with a single subnet prefix shared among the MANET nodes implies that all MANET nodes would be considered as belonging to the same subnet – and as such on the same IP link. However with the MANET forming a multi-hop L3 network,, L3 forwarding of IP datagrams may occur, and with such forwarding, TTL/hop-limit are decremented; link-local multicast or broadcasts either do not reach all nodes within the subnet – or if they are to reach all nodes within the subnet, they are to be forwarded by intermediate nodes. In short, considering and configuring MANET nodes as if the MANET forms a single subnet breaks the classic IP link model and the applications which assume the characteristics of the classic IP link model. [4] explores this in more detail.

2.6.1 Routing Incompatibility

A perhaps surprising example of an application, which breaks under this common MANET misperception, is routing: if a multi-hop MANET is configured as described in this section,

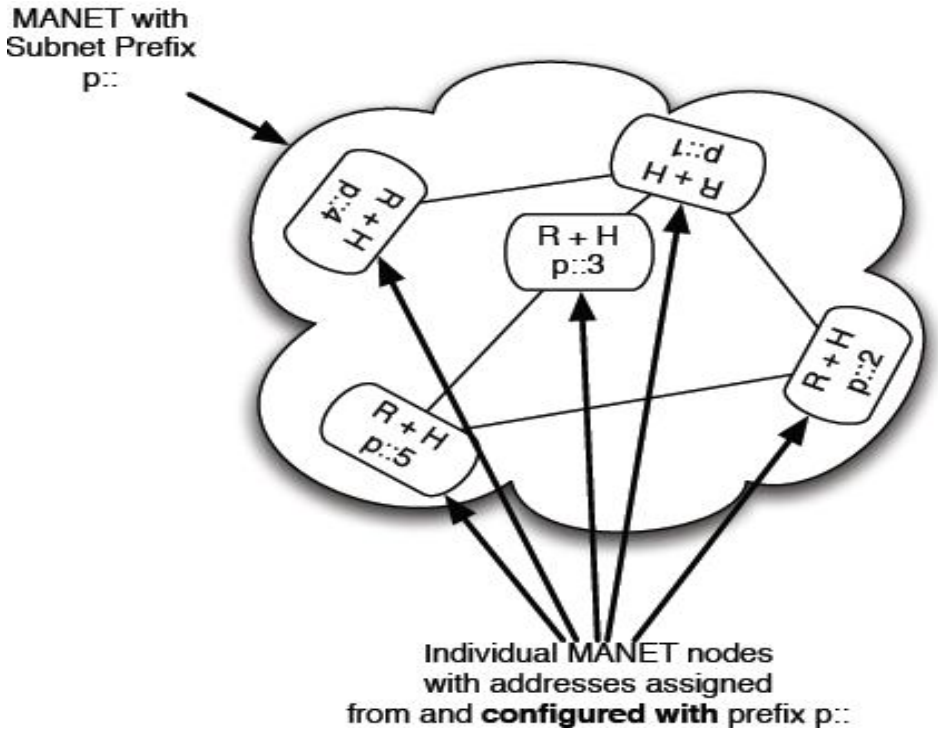


Figure 2.7: Common misperception of a MANET: viewing the MANET as a classic IP subnet

With all nodes within the MANET assumed to be also in the same subnet, then forwarding of IP datagrams within the MANET will prompt intermediate nodes to produce ICMP redirects [25]. This is appropriate since IP datagrams delivered within a subnet are not supposed to be forwarded by a router since a direct link between any two nodes within a subnet is supposed to exist, according to the classic IP link model.

A rough work-around, often proposed in order to ”mask” this problem, is to disable ICMP redirect.

2.6.2 Incompatibility with Other Protocols and Applications

Disabling ICMP redirects to make routing operate is disabling the symptom of an incorrect network model, for a single application (routing) only, and leads to the specific and reasonable question if other applications and protocols require similar tweaks (if so, which applications/protocols and which tweaks?). Even more general: one could ask if MANETs even do belong in the IP world? [12]The answer is yes, MANETs do belong in the IP world –however it also means that the architectural view, presented in this section, is inappropriate and indeed a common misperception of MANETs, which does not take into consideration their integration within the IP architecture.

2.7 A MANET Architectural Model

This section presents an architectural model for MANETs which preserves the integrity of the IP architecture while allowing for the particularities of MANETs.

2.7.1 MANET Node Morphology

This architectural model considers MANET nodes as routers with hosts attached, as illustrated in figure 8. These attached hosts may be ”external” (i.e. attached to the router via other network interfaces) or ”internal” – however the important observation to make is, that the links

between these hosts and the router are classic IP links [26]. This implies that, from the point of view of the hosts, and the applications running on these hosts, connectivity is via a classic IP link. Hosts, and their applications, are not exposed to the specific characteristics of the MANET interfaces and are connected to the MANET via a router, which has one or more MANET interfaces. This is symmetric with how hosts on an Ethernet, such as illustrated in Figure 2.8 are not exposed to the intricacies of what type of connectivity the router has beyond the Ethernet.

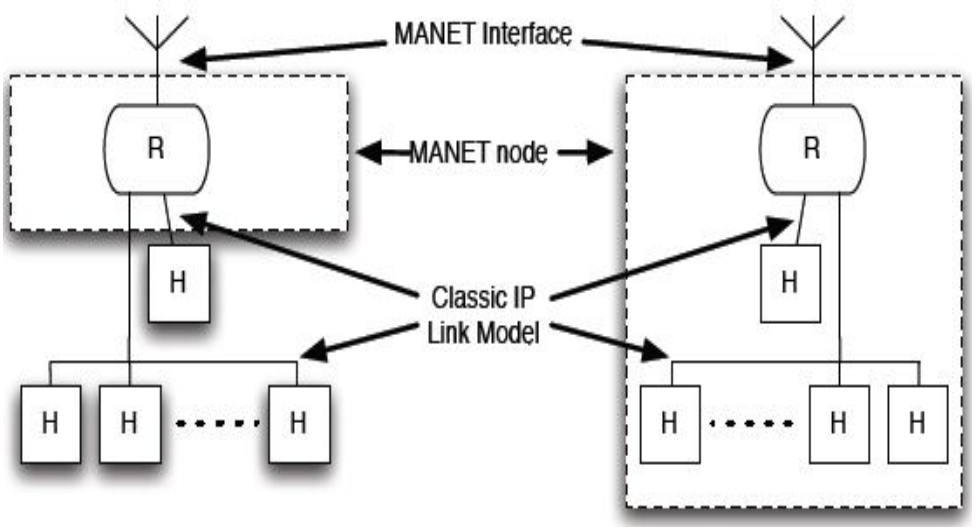


Figure 2.8: MANET node model: the router (R) has on the top a MANET interface, and is connected, on the bottom, to hosts (H) via classic IP links.

Since the hosts in Figure 2.8 are connected to a classic IP link, these hosts are configured and behave as hosts in any other network, and the links to which they are connected have properties identical to those of any other classic IP link.

2.7.2 Addresses and Prefixes

If the MANET router is delegated a prefix p::, this prefix can be assigned to the classic IP link(s), and hosts can be assigned addresses from within this prefix, and configured with this prefix as illustrated in Figure 2.8. Specifically, the MANET interface(s) of the router are not configured with this prefix, [27]: the MANET interface(s) is not on the same ”link” as the other interfaces with addresses from within this prefix, and so direct communication without crossing a router is not possible. The configuration of MANET interfaces is detailed below.

2.7.3 MANET Interface Configuration & Properties

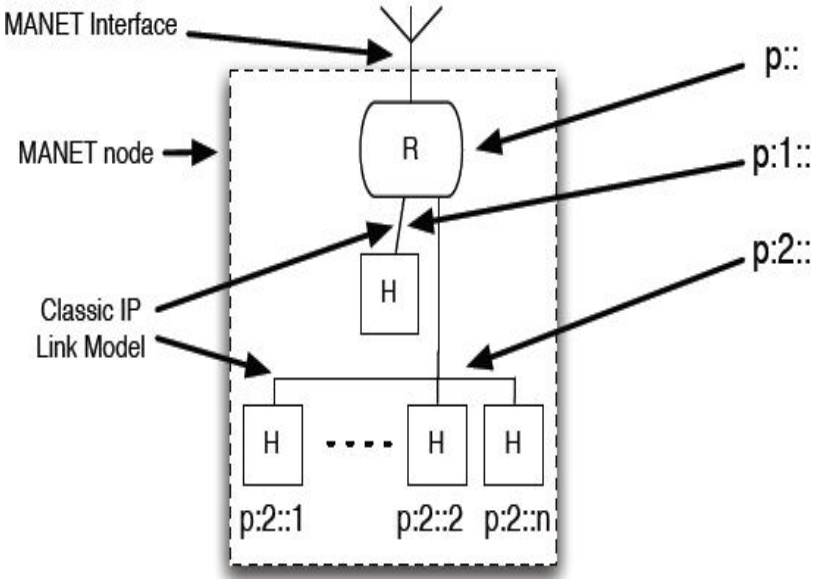


Figure 2.9: MANET node and prefixes: the MANET router (R) is delegated a prefix p::, which it assigns to the classic IP links to which the hosts (H) are attached.

MANET specific behaviors are exclusively exposed to the MANET interface(s) of the routers [28]. In Figure 2.9 the scenario includes MANET routing protocols and interface and link characteristics (asymmetric

neighborhoods, semi-broadcast interfaces, fuzzy neighbor relationships, topology dynamics etc.) The following characteristics deserve particular mention, since they distinguish MANET interfaces and the MANET link model from the classic IP link model:

Unique Prefixes

MANET interfaces must be configured with unique prefixes, i.e. such that no two MANET interfaces are configured such that they appear within the same IP subnet [19].

Some common ways to achieve this are:

- unnumbered interfaces (IPv4);
- Link-Local Addresses (IPv6);
- /128 (IPv6) or /32 (IPv4) prefixes.

However it is worth noting that prefix lengths shorter than /128 (IPv6) or /32 (IPv4) are possible on the MANET interface, so long as the prefixes are unique to a single MANET interface.

Link Local Mulitcast/broadcast Scope

On a MANET interface, a Link Local multicast or broadcasts reach MANET interfaces of neighbor nodes only, regardless of their configured addresses. A Link Local multicast or broadcast on a MANET interface is, thus, a ”neighbor cast”, and is not forwarded nor assumed to be received by all nodes within a MANET [29].

2.7.4 MANET Network View

Following the architecture described in previously, a configured MANET with routers and hosts, looks as in Figure 2.10:

- the inner white cloud represents where MANET interfaces and links form a MANET –
- and the outer gray cloud represents where the classic IP link model is assumed [30].

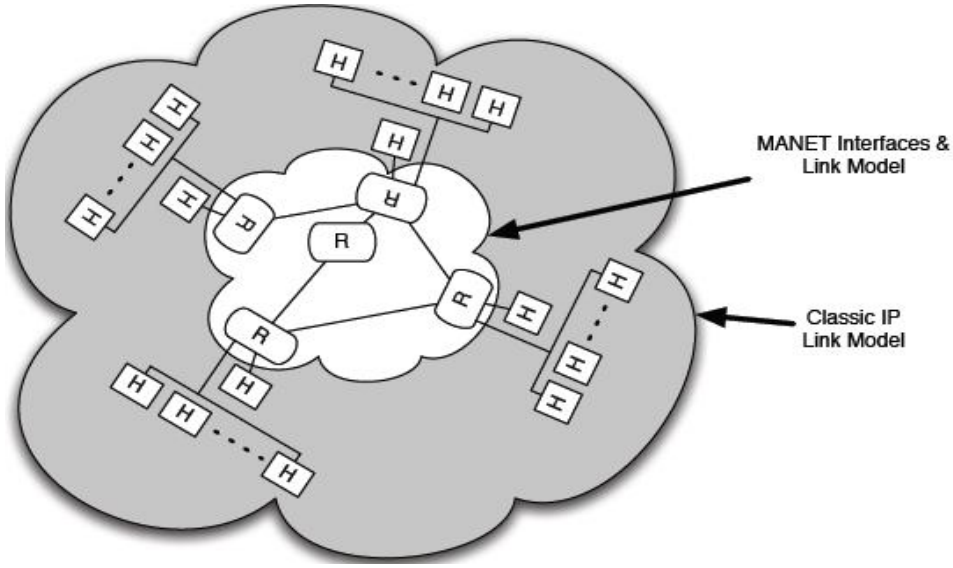


Figure 2.10: MANET Network Model: the inner white cloud is where MANET interfaces and links for a MANET are found and MANET specific protocols apply. The outer gray cloud represents where the classic IP link model (and regular applications/protocols) applies.

Chapter 3 MANETs Security

This chapter provides an overview of MANETs security, concentrating the focus on the key concepts these are relevant to the security of MANETs. It is organized into six subsections demonstrating the types of attacks that could happened in MANETs, then providing the set of security services available to MANETs, finally the last two subsections gives insights to the two types of encryption used through the research.

3.1 Review of MANETs Security

MANETs Information security may be defined as is the practice of defending information from unauthorized access, use, disclosure, disruption, modification, perusal, inspection, recording or destruction. It is a general term that can be used regardless of the form the data may take**.** The requirements of information security within an organization have undergone two major changes in the last several decades. Before the widespread use of data processing equipment, the security of information felt to be valuable to an organization was provided primarily by physical and administrative means. An example of the former is the use of rugged filing cabinets with a combination lock for storing sensitive documents. An example of the latter is personnel screening procedures used during the hiring process [31].

With the introduction of the computer, the need for automated tools for protecting files and other information stored on the computer became evident. This is especially the case for a shared system, such as a time-sharing system, and the need is even more acute for systems that can be accessed over a public telephone network, data network, or the Internet. The

generic name for the collection of tools designed to protect data and to thwart hackers is **computer security**.

The second major change that affected security is the introduction of distributed systems and the use of networks and communications facilities for carrying data between terminal user and computer and between computer and computer. **Network security** measures are needed to protect data during their transmission. In fact, the term network security is somewhat misleading, because virtually all business, government, and academic organizations interconnect their data processing equipment with a collection of interconnected networks. Such a collection is often referred to as an internet, and the term **internet security** is used [32].

We will start by introducing some important terms before going through with te discussion of security:

- **Security attack:** Any action that compromises the security of information owned by an organization.
- **Security mechanism:** A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.
- **Security service:** A processing or communication service that enhances the security of the data processing systems and the information transfers of an organization. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.
- **Threat:** A potential for violation of security, which exists when there is a circumstance, capability, action, or event that could breach security and cause harm. That is, a threat is a possible danger that might exploit a vulnerability.

- **Attack:** An assault on system security that derives from an intelligent threat; that is, an intelligent act that is a deliberate attempt (especially in the sense of a method or technique) to evade security services and violate the security policy of a system.

3.2 Security Attacks Types:

A useful means of classifying security attacks, used both in X.800 and RFC 2828, is in terms of *passive attacks* and *active attacks*. A passive attack attempts to learn or make use of information from the system but does not affect system resources. An active attack attempts to alter system resources or affect their operation.

3.2.1 Passive Attacks

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are release of message contents and traffic analysis [33].

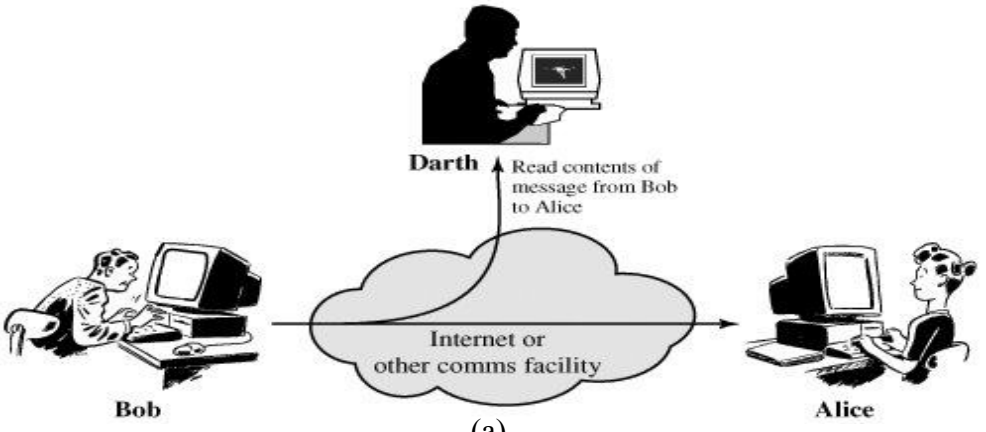
The **release of message contents** is easily understood A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions.

A second type of passive attack, **traffic analysis**, is subtler (Figure 3.1b). Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption.

If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place[34].

Passive attacks are very difficult to detect because they do not involve any alteration of the data.

Typically, the message traffic is sent and received in an apparently normal fashion and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection. Figure 3.1 illustrates the types of passive attacks [32]



(a)

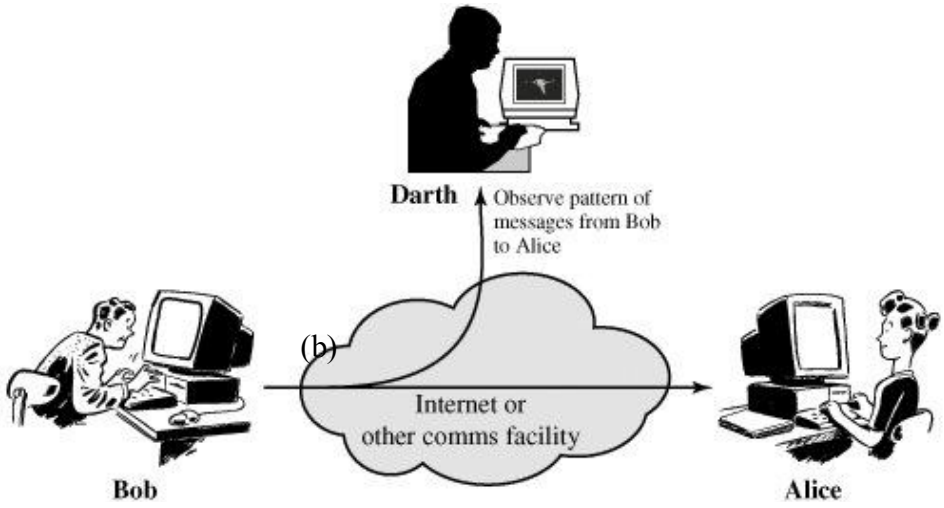


Figure 3.1 types of passive attacks (a) Release of message contents (b) Traffic analysis[30]

3.2.2 Active Attacks

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

3.2.2.1 masquerade attack

This type of attacks takes place when one entity pretends to be a different entity (Figure 3.2) [34]. A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorized entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

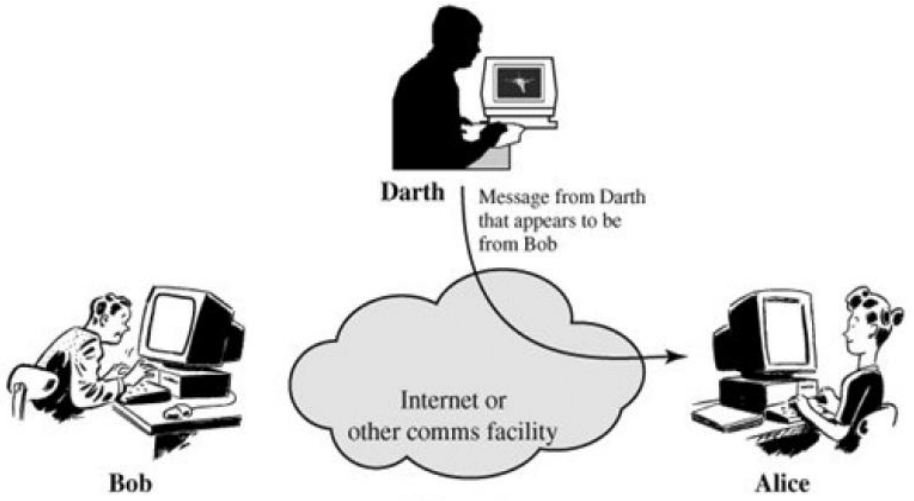


Figure 3.2 masquerade attack

3.2.2.2Replay attack

It involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorized effect (Figure 3.3).

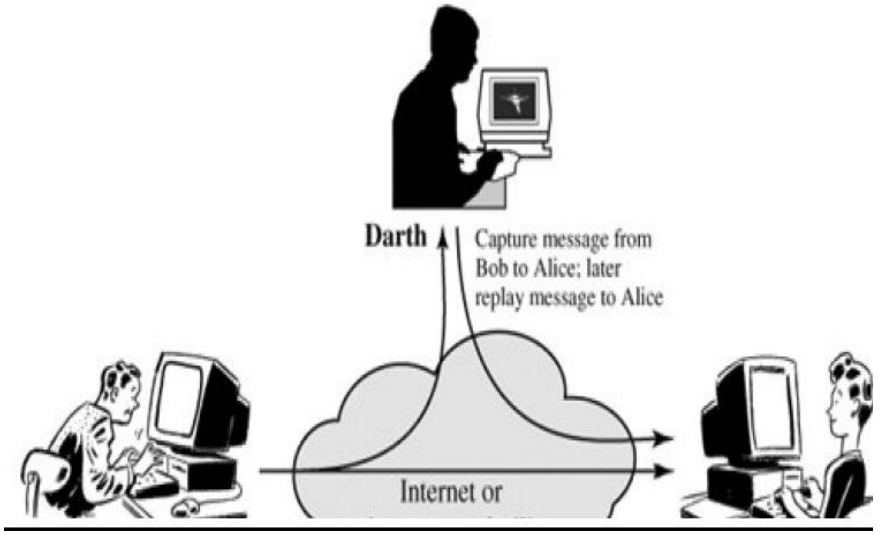


Figure 3.3 Reply attack

3.2.2.3Modification of messages

That simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorized effect (Figure 3.4). For example, a message meaning [35] "Allow John Smith to read confidential file *accounts*" is modified to mean "Allow Fred Brown to read confidential file *accounts.*"

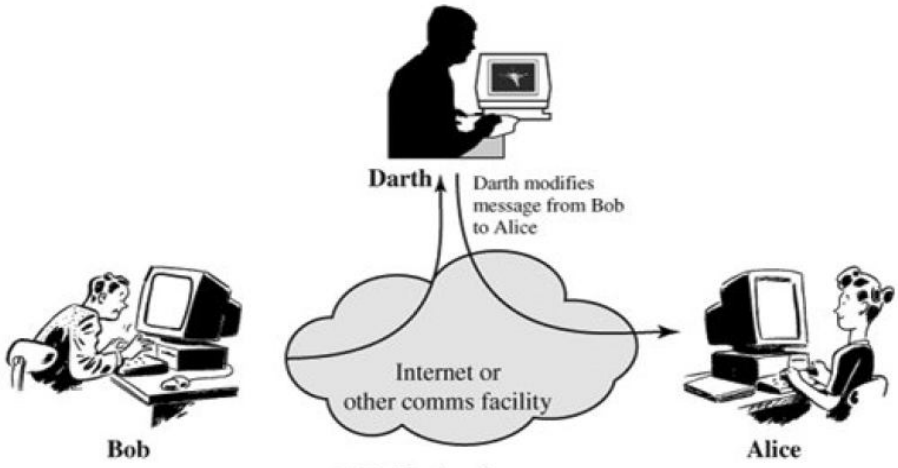


Figure 3.4 modification attack

3.2.2.4 Denial of Service

prevents or inhibits the normal use or management of communications facilities as in Figure 3.5. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination (e.g., the security audit service). Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance[36].

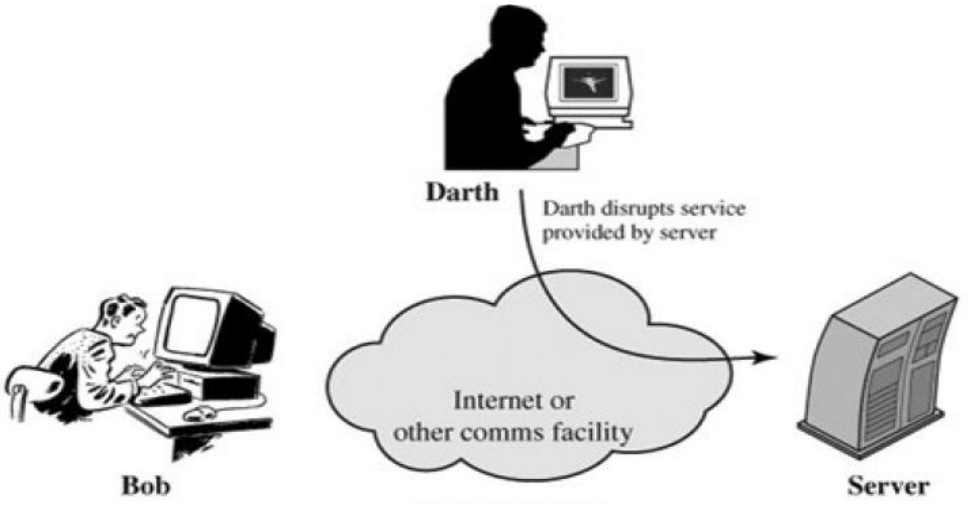


Figure 3.5 DoS attack

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks absolutely, because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention [37].

3.3 MANETs Security Services

X.800 defines a security service as a service provided by a protocol layer of communicating open systems, which ensures adequate security of the systems or of data transfers [28]. Perhaps a clearer definition is found in RFC 2828, which provides the following definition: a processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms.

3.3.1 AUTHENTICATION

The assurance that that the data, transactions, communications or documents (electronic or physical) are genuine. It is also important for authenticity to validate that both parties involved are who they claim to be. Some information security systems incorporate authentication features such as "digital signatures", which give evidence that the message data is genuine and was sent by someone possessing the proper signing key[38].

- **Peer Entity Authentication:** Used in association with a logical connection to provide confidence in the identity of the entities connected.
- **Data Origin Authentication:** In a connectionless transfer, provides assurance that the source of received data is as claimed.

3.3.2 ACCESS CONTROL

Access to protected information must be restricted to people who are authorized to access the information. The computer programs, and in many cases the computers that process the information, must also be authorized. This requires that mechanisms be in place to control the access to protected information. The sophistication of the access control mechanisms should be in parity with the value of the information being protected – the more sensitive or valuable the information the stronger the control mechanisms need to be [25]]. it is also may be seen as the prevention of unauthorized use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).

3.3.3 DATA CONFIDENTIALITY

Confidentiality refers to preventing the disclosure of information to unauthorized individuals or systems. For example, a credit card transaction on the Internet requires the credit card number to be transmitted from the buyer to the merchant and from the merchant to a transaction processing network. The system attempts to enforce confidentiality by encrypting the card number during transmission, by limiting the places where it might appear (in databases, log files, backups, printed receipts, and so on), and by restricting access to the places where it is stored. If an unauthorized party obtains the card number in any way, a breach of confidentiality has occurred [39].

Confidentiality is necessary for maintaining the privacy of the people whose personal information a system holds..

- **Connection Confidentiality:** The protection of all user data on a connection.
- **Connectionless Confidentiality:** The protection of all user data in a single data block
- **Selective-Field Confidentiality:** The confidentiality of selected fields within the user data on a connection or in a single data block.
- **Traffic Flow Confidentiality:** The protection of the information that might be derived from observation of traffic flows.

3.3.4 DATA INTEGRITY data integrity means maintaining and assuring the accuracy and consistency of data over its entire life-cycle.[7] This means that data cannot be modified in an unauthorized or undetected manner. This

is not the same thing as referential integrity in databases, although it can be viewed as a special case of Consistency as understood in the classic ACID model of transaction processing. Integrity is violated when a message is actively modified in transit. Information security systems typically provide message integrity in addition to data confidentiality [40].

- **Connection Integrity with Recovery:** Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
- **Connection Integrity without Recovery:** As above, but provides only detection without recovery.
- **Selective-Field Connection Integrity:** Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.
- **Connectionless Integrity:** Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
- **Selective-Field Connectionless Integrity:** Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified [41].

3.3.5 NONREPUDIATION

Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication.

In law, non-repudiation implies one's intention to fulfill their obligations to a contract. It also implies that one party of a transaction cannot deny having received a transaction nor can the other party deny having sent a transaction It is important to note that while technology such as cryptographic systems can assist in non-repudiation efforts, the concept is at its core a legal concept transcending the realm of technology [42]. It is not, for instance, sufficient to show that the message matches a digital signature signed with the sender's private key, and thus only the sender could have sent the message and nobody else could have altered it in transit. The alleged sender could in return demonstrate that the digital signature algorithm is vulnerable or flawed, or allege or prove that his signing key has been compromised. The fault for these violations may or may not lie with the sender himself, and such assertions may or may not relieve the sender of liability, but the assertion would invalidate the claim that the signature necessarily proves authenticity and integrity and thus prevents repudiation.

- **Non-repudiation, Origin:** Proof that the message was sent by the specified party.
- **Non-repudiation, Destination:** Proof that the message was received by the specified party.

3.4 MANETs Security Mechanisms

3.4.1 SPECIFIC SECURITY MECHANISMS:

May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services [43].

- **Encipherment:** The use of mathematical algorithms to transform data into a form that is not readily intelligible. The

transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.

- **Digital Signature:** Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient).
- **Access Control:** A variety of mechanisms that enforce access rights to resources.
- **Data Integrity:** A variety of mechanisms used to assure the integrity of a data unit or stream of data units.
- **Authentication Exchange:** A mechanism intended to ensure the identity of an entity by means of information exchange.
- **Traffic Padding:** The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.
- **Routing Control:** Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.
- **Notarization:** The use of a trusted third party to assure certain properties of a data exchange.

3.4.2 PERVASIVE SECURITY MECHANISMS:

Mechanisms that are not specific to any particular OSI security service or protocol layer [44].

- **Trusted Functionality:** That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).

- **Security Label:** The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
- **Event Detection:** Detection of security-relevant events.
- **Security Audit Trail:** Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
- **Security Recovery:** Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

3.5 A Model for Network Security

A model for much of what we will be discussing is captured, in very general terms, in Figure 3.6. A message is to be transferred from one party to another across some sort of internet [45]. The two parties, who are the *principals* in this transaction, must cooperate for the exchange to take place. A logical information channel is established by defining a route through the internet from source to destination and by the cooperative use of communication protocols (e.g., TCP/IP) by the two principals.

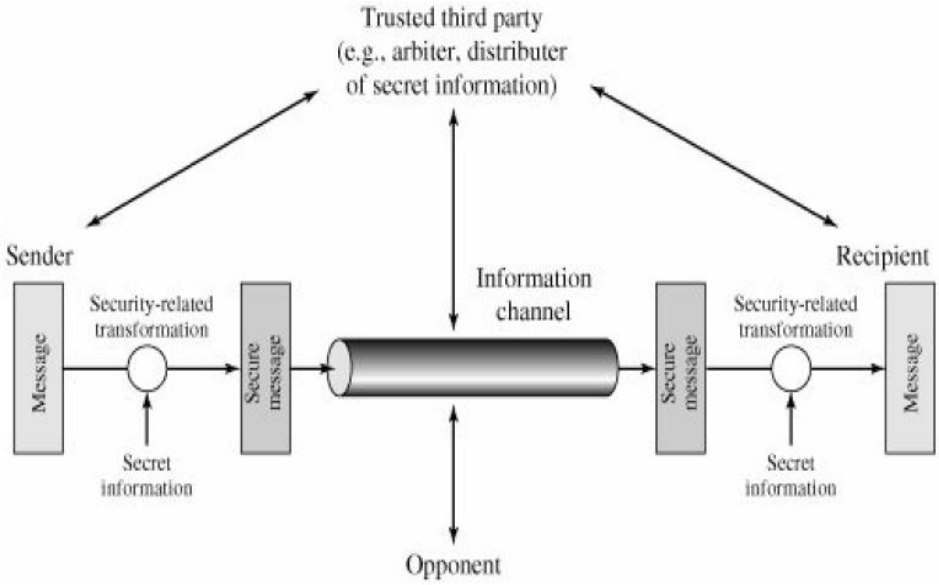


Figure 3.6 General model for security

Security aspects come into play when it is necessary or desirable to protect the information transmission from an opponent who may present a threat to confidentiality, authenticity, and so on. All the techniques for providing security have two components [46]:

- A security-related transformation on the information to be sent. Examples include the encryption of the message, which scrambles the message so that it is unreadable by the opponent, and the addition of a code based on the contents of the message, which can be used to verify the identity of the sender
- Some secret information shared by the two principals and, it is hoped, unknown to the opponent. An example is an encryption key used in conjunction with the transformation to scramble the message before transmission and unscramble it on reception.

3.6 Encryption techniques

In this subsection an overview for the encryption techniques is provided, starting from the symmetric encryption then going to the asymmetric encryption which is the main focus of this research.

3.6.1 Symmetric Encryption model

A symmetric encryption scheme has five ingredients as shown in Figure 3.7:

- **Plaintext:** This is the original intelligible message or data that is fed into the algorithm as input.
- **Encryption algorithm:** The encryption algorithm performs various substitutions and transformations on the plaintext.
- **Secret key:** The secret key is also input to the encryption algorithm. The key is a value independent of the plaintext and of the algorithm. The algorithm will produce a different output depending on the specific key being used at the time. The exact substitutions and transformations performed by the algorithm depend on the key [47].
- **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the secret key. For a given message, two different keys will produce two different ciphertexts. The ciphertext is an apparently random stream of data and, as it stands, is unintelligible.

- **Decryption algorithm:** This is essentially the encryption algorithm run in reverse. It takes the ciphertext and the secret key and produces the original plaintext [48].

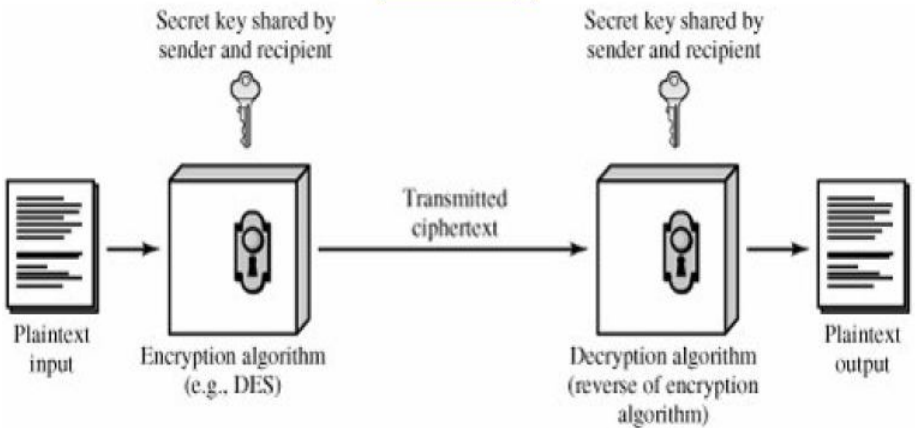


Figure 3.7 Symmetric encryption

3.6.2 Substitution Encryption Techniques:

In this section, we examine a sampling of what might be called classical encryption techniques. A study of these techniques enables us to illustrate the basic approaches to symmetric encryption used today and the types of cryptanalytic attacks that must be anticipated.

3.6.2.1 Caesar Cipher

In cryptography, a Caesar cipher, also known as Caesar's cipher, the shift cipher, Caesar's code or Caesar shift, is one of the simplest and most widely known encryption techniques. It is a type of substitution cipher in which each letter in the plaintext is replaced by a letter some fixed number of positions down the alphabet [40]. For example, with a left shift of 3, D would be replaced by A, E would

become B, and so on. The method is named after Julius Caesar, who used it in his private correspondence. For example,

```
plain: meet me after the toga party
cipher: PHHW PH DIWHU WKH WRJD SDUWB
```

Note that the alphabet is wrapped around, so that the letter following Z is A. We can define the transformation by listing all possibilities, as follows:

```
plain: a b c d e f g h i j k l m n o p q r s t u v w x y z
cipher: D E F G H I J K L M N O P Q R S T U V W X Y Z A B C
```

Let us assign a numerical equivalent to each letter. Then the algorithm can be expressed as follows. For each plaintext letter p, substitute the ciphertext letter C:

$$C = E(k, p) = (p + k) \bmod 26$$

where k takes on a value in the range 1 to 25. The decryption algorithm is simply

$$p = D(k, C) = (C\ k) \bmod 26$$

3.6.2.2 Monoalphabetic Ciphers

The mono-alphabetic substitution cipher is so called because each plain text letter is substituted by the same cipher text letter throughout the entire message [49].With only 25 possible keys, the Caesar cipher is far from secure. A dramatic increase in the key space can be achieved by allowing an arbitrary substitution. the "cipher" line can be any permutation of the 26 alphabetic characters, then there are 26! or greater than 4 x 1026 possible keys. This is 10 orders of magnitude greater than

the key space for DES and would seem to eliminate brute-force techniques for cryptanalysis. Such an approach is referred to as a **monoalphabetic substitution cipher**, because a single cipher alphabet (mapping from plain alphabet to cipher alphabet) is used per message.

There is, however, another line of attack. If the cryptanalyst knows the nature of the plaintext (e.g., noncompressed English text), then the analyst can exploit the regularities of the language.

3.6.2.3 Playfair Cipher

The **Playfair cipher** or **Playfair square** is a manual symmetric encryption technique and was the first literal digraph substitution cipher. The scheme was invented in 1854 by Charles Wheatstone, but bears the name of Lord Playfair who promoted the use of the cipher [50].

The technique encrypts pairs of letters (*digraphs*), instead of single letters as in the simple substitution cipher and rather more complex Vigenère cipher systems then in use.

The Playfair is thus significantly harder to break since the frequency analysis used for simple substitution ciphers does not work with it. Frequency analysis can still be undertaken, but on the $600^{[1]}$ possible digraphs rather than the 26 possible monographs. The frequency analysis of digraphs is possible, but considerably more difficult – and it generally requires a much larger ciphertext in order to be useful. .it is the best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams. In this case, the keyword is *monarchy*. The matrix is constructed by filling in the letters of the

keyword (minus duplicates) from left to right and from top to bottom, and then filling in the remainder of the matrix with the remaining letters in alphabetic order. The letters I and J count as one letter. Plaintext is encrypted two letters at a time, according to the following rules:

1. Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.

2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.

3. Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.

4. Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

E	G	P	T	F
R	A	B	C	D
H	I/J	K	L	M
N	O	Q	S	U
V	W	X	Y	Z

Figure 3.8 playfiar cipher example [50].

The Playfair cipher (illustrated in figure 3.8) is a great advance over simple monoalphabetic ciphers. For one thing, whereas there are only 26 letters, there are 26 x 26 = 676 digrams, so that identification of individual digrams is more difficult. Furthermore, the relative frequencies of individual letters exhibit a much greater range than that of digrams, making frequency analysis much more difficult. For these reasons, the Playfair cipherwas for a long time considered unbreakable. It was used as the standard field system by the British army in World War I and still enjoyed considerable use by the U.S. Army and other Allied forces during World War II.

Despite this level of confidence in its security, the Playfair cipher is relatively easy to break because it still leaves much of the structure of the plaintext language intact. A few hundred letters of ciphertext are generally sufficient.

3.6.2.4 Polyalphabetic Ciphers

A polyalphabetic cipher is any cipher based on substitution, using multiple substitution alphabets [51]. The Vigenère cipher is probably the best-known example of a polyalphabetic cipher, though it is a simplified special case. The Enigma machine is more complex but still fundamentally a

polyalphabetic substitution cipher. This type of ciphers is another way to improve on the simple monoalphabetic technique is to use different monoalphabetic substitutions as one proceeds through the plaintext message. The general name for this approach is polyalphabetic substitution cipher. All these techniques have the following features in common:

1. A set of related monoalphabetic substitution rules is used.
2. A key determines which particular rule is chosen for a given transformation.

The best known, and one of the simplest, such algorithm is referred to as the Vigenère cipher. In this scheme, the set of related monoalphabetic substitution rules consists of the 26 Caesar ciphers, with shifts of 0 through 25. Each cipher is denoted by a key letter, which is the ciphertext letter that substitutes for the plaintext letter a. Thus, a Caesar cipher with a shift of 3 is denoted by the key value d like in figure 3.9.

Key \ Plaintext	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z
a	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
b	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A
c	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B
d	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C
e	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D
f	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E
g	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F
h	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G
i	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H
j	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I
k	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J
l	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K
m	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L
n	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M
o	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N
p	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
q	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
r	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
s	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R
t	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S
u	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T
v	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
w	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V
x	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
y	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
z	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y

Figure 3.9 Polyalphabetic cipher example

To encrypt a message, a key is needed that is as long as the message. Usually, the key is a repeating keyword. For example, if the keyword

is *deceptive*, the message "we are discovered save yourself" is encrypted as follows:

```
key:            deceptivedeceptivedeceptive
plaintext:      wearediscoveredsaveyourself
ciphertext:     ZICVTWQNGRZGVTWAVZHCQYGLMGJ
```

Decryption is equally simple. The key letter again identifies the row. The position of the ciphertext letter in that row determines the column, and the plaintext letter is at the top of that column.

The strength of this cipher is that there are multiple ciphertext letters for each plaintext letter, one for each unique letter of the keyword. Thus, the letter frequency information is obscured. However, not all knowledge of the plaintext structure is lost. An improvement is achieved over the Playfair cipher, but considerable frequency information remains.

3.6.3 Asymmetric Encryption:

3.6.3.1Principles of Public-Key Cryptosystems

Public-key cryptography, also known as asymmetric cryptography, refers to a cryptographic algorithm which requires two separate keys one of which is secret (or private) and one of which is public [52]. Although different, the two parts of this key pair are mathematically linked. The public key is used to encrypt plaintext or to verify a digital signature; whereas the private key is used to decrypt ciphertext or to create a digital signature. The term "asymmetric" stems from the use of different keys to perform these opposite functions, each the inverse of the other – as contrasted with

conventional ("symmetric") cryptography which relies on the same key to perform both.

Public-key algorithms are based on mathematical problems which currently admit no efficient solution that are inherent in certain integer factorization, discrete logarithm, and elliptic curve relationships. It is computationally easy for a user to generate their public and private key-pair and to use them for encryption and decryption. The strength lies in the fact that it is "impossible" (computationally infeasible) for a properly generated private key to be determined from its corresponding public key. Thus the public key may be published without compromising security, whereas the private key must not be revealed to anyone not authorized to read messages or perform digital signatures. Public key algorithms, unlike symmetric key algorithms, do not require a secure initial exchange of one (or more) secret keys between the parties [53]. The concept of public-key cryptography evolved from an attempt to attack two of the most difficult problems associated with symmetric encryption. The first problem is that of key distribution.

As it is obvious, key distribution under symmetric encryption requires either

(1) that two communicants already share a key, which somehow has been distributed to them; or

(2) the use of a key distribution center. Whitfield Diffie, one of the discoverers of public-key encryption (along with Martin Hellman, both at Stanford University at the time), reasoned that this second requirement negated the very essence of cryptography: the ability to maintain total secrecy over your own communication.

As Diffie put it [DIFF88], "what good would it do after all to develop impenetrable cryptosystems, if their users were forced to share their keys

with a KDC that could be compromised by either burglary or subpoena?. The second problem that Diffie pondered, and one that was apparently unrelated to the first was that of "digital signatures." If the use of cryptography was to become widespread, not just in military situations but for commercial and private purposes, then electronic messages and documents would need the equivalent of signatures used in paper documents [54]. That is, could a method be devised that would stipulate, to the satisfaction of all parties, that a digital message had been sent by a particular person?

A public-key encryption scheme has six ingredients (Figure 3.10a):

a) **Plaintext:** This is the readable message or data that is fed into the algorithm as input.
b) **Encryption algorithm:** The encryption algorithm performs various transformations on the plaintext.
c) **Public and private keys:** This is a pair of keys that have been selected so that if one is used for encryption, the other is used for decryption. The exact transformations performed by the algorithm depend on the public or private key that is provided as input.
d) **Ciphertext:** This is the scrambled message produced as output. It depends on the plaintext and the key. For a given message, two different keys will produce two different ciphertexts.
e) **Decryption algorithm:** This algorithm accepts the ciphertext and the matching key and produces the original plaintext.

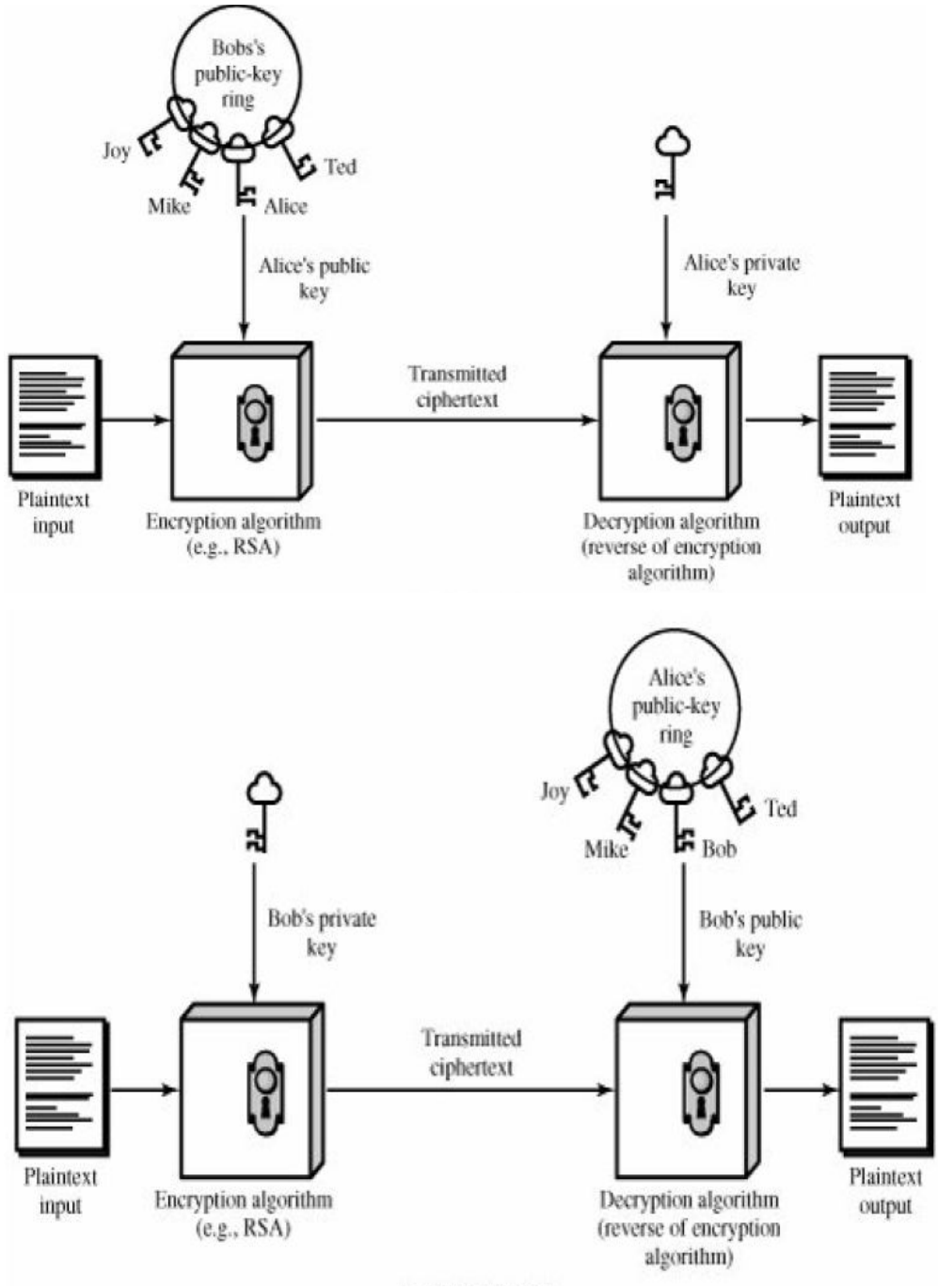


Figure 3.10:Asymmetric Key operations (a)Encryption (b) Authentication

The essential steps for asymmetric encryption are:

1. Each user generates a pair of keys to be used for the encryption and decryption of messages.

2. Each user places one of the two keys in a public register or other accessible file. This is the public key. The companion key is kept private. As Figure 3.10a suggests, each user maintains a collection of public keys obtained from others.

3. If Bob wishes to send a confidential message to Alice, Bob encrypts the message using Alice's public key.

4. When Alice receives the message, she decrypts it using her private key. No other recipient can decrypt the message because only Alice knows Alice's private key.

With this approach, all participants have access to public keys, and private keys are generated locally by each participant and therefore need never be distributed. As long as a user's private key remains protected and secret, incoming communication is secure [55]. At any time, a system can change its private key and publish the companion public key to replace its old public key.

Table 3.1 provides a brief comparison between public key and conventional encryption

Table 3.1 comparing symmetric to asymmetric encryption

	Conventional Encryption	**Public-Key Encryption**
Needed to Work	**1.** The same algorithm with the same key is used for encryption and decryption [56]. **2.** The sender and receiver must share the algorithm and the key .	1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption [57]. **2.** The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security	1. The key must be kept secret. 2. It must be impossible or at least impractical to decipher a message if no other information is available [58]. 3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key [60].	**1.** One of the two keys must be kept secret. **2.** It must be impossible or at least impractical to decipher a message if no other information is available [59]. **3.** Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key [61].

3.6.3.2Requirements for Public-Key Cryptography

The cryptosystem illustrated in Figures 3.11 and 3.12 depends on a cryptographic algorithm based on two related keys. Diffie and Hellman postulated this system without demonstrating that such algorithms exist. However, they did lay out the conditions that such algorithms must fulfill [DIFF76b] [62]:

1. It is computationally easy for a party B to generate a pair (public key *PUb*, private key *PRb*).

2. It is computationally easy for a sender A, knowing the public key and the message to be encrypted, *M*, to generate the corresponding ciphertext: $C = E(PUb, M)$

3. It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message: $M = D(PRb, C) = D[PRb, E(PUb, M)]$

4. It is computationally infeasible for an adversary, knowing the public key, *PUb*, to determine the private key, *PRb*.

5. It is computationally infeasible for an adversary, knowing the public key, *PUb*, and a ciphertext, *C*, to recover the original message, *M*.

6. The two keys can be applied in either order:

$$M = D[PU_b, E(PR_b, M)] = D[PR_b, ME(PU_b, M)]$$

3.6.3.3 The RSA Algorithm

RSA is an algorithm for public-key cryptography that is based on the presumed difficulty of factoring large integers, the factoring problem. RSA stands for Ron Rivest, Adi Shamir and Leonard Adleman, who first publicly described the algorithm in 1977. Clifford Cocks, an English mathematician,

had developed an equivalent system in 1973, but it wasn't declassified until 1997.[63]

A user of RSA creates and then publishes the product of two large prime numbers, along with an auxiliary value, as their public key. The prime factors must be kept secret. Anyone can use the public key to encrypt a message, but with currently published methods, if the public key is large enough, only someone with knowledge of the prime factors can feasibly decode the message.[64] Whether breaking RSA encryption is as hard as factoring is an open question known as the RSA problem. The RSA scheme is a block cipher in which the plaintext and ciphertext are integers between 0 and n 1 for some n. A typical size for n is 1024 bits, or 309 decimal digits. That is, n is less than 21024. We examine RSA in this section in some detail, beginning with an explanation of the algorithm. Then we examine some of the computational and cryptanalytical implications of RSA.

The scheme developed by Rivest, Shamir, and Adleman makes use of an expression with exponentials [65].

Plaintext is encrypted in blocks, with each block having a binary value less than some number n. That is, the block size must be less than or equal to log2(n); in practice, the block size is i bits, where $2^i < n \leq 2^{i+1}$.

Encryption and decryption are of the following form, for some plaintext block M and ciphertext block C: $C = M^e \bmod n$

$$M = C^d \bmod n = (M^e)^d \bmod n = M^{ed} \bmod n$$

Key Generation

Select p, q	p and q both prime, $p \neq q$
Calculate $n = p \times q$	
Calculate $\phi(n) = (p - 1)(q - 1)$	
Select integer e	$\gcd(\phi(n), e) = 1; 1 < e < \phi(n)$
Calculate d	$d \equiv e^{-1} \pmod{\phi(n)}$
Public key	$PU = \{e, n\}$
Private key	$PR = \{d, n\}$

Figure 3.11Key generation for RSA

Encryption

Plaintext:	$M < n$
Ciphertext:	$C = M^e \bmod n$

Decryption

Ciphertext:	C
Plaintext:	$M = C^d \bmod n$

Figure 3.12: Encryption and Decryption in RSA

Chapter 4: The proposed Key Distribution Model for Botnets Prevention in MANETs

This chapter is devoted to describe the dangerous effects of Botnets in MANETs and how to stand against bots and bots' controllers in an open environment such as MANETs this chapter is divided into two main sub sections the first 4.1 provides review for the Botnets to illustrate the meaning and the harmful effects for this type of attack. In section 4.2 we provide a method to stand against Botnets via distributing session keys in advance to every mobile unit registering its self to the MANETs log directory.

4.1 Overview of Botnets

Botnet is a jargon term for a collection of software agents, or robots, that run autonomously and automatically. The term is most commonly associated with malicious software, but it can also refer to the network of computers using distributed computing software [61]. While Botnets are often named after their malicious software name, there are typically multiple Botnets in operation using the same malicious software families, but operated by different criminal entities

Botnets have become a significant part of the Internet, albeit increasingly hidden. Due to most conventional Internet Rely Chat (IRC) networks taking measures and blocking access to previously-hosted Botnets, controllers must now find their own servers [62]. Often, a Botnet will include a variety of connections and network types. Sometimes a controller will hide an IRC server installation on an educational or corporate site where high-speed

connections can support a large number of other bots. Exploitation of this method of using a bot to host other bots has proliferated only recently as most script kiddies do not have the knowledge to take advantage of it.

While the term "Botnet" can be used to refer to any group of bots, such as IRC bots, this word is generally used to refer to a collection of compromised computers (called zombie computers) running software, usually installed via drive-by downloads exploiting web browser vulnerabilities, worms, Trojan horses, or backdoors, under a common command-and-control infrastructure [63].The untraceable feature of coordinated attacks is just what hackers/attackers demand to compromise a computer or a network for their illegal activities. Once a group of hosts at different locations controlled by a malicious individual or organization to initiate an attack, one can hardly trace back to the origin due to the complexity of the Internet. For this reason, the increase of events and threats against legitimate Internet activities such as information leakage, click fraud, denial of service (DoS) and attack, E-mail spam, etc., has become a very serious problem nowadays. Those victims controlled by coordinated attackers are called zombies or bots which derives from the word “robot.” The term of bots is commonly referred to software applications running as an automated task over the Internet [64]. Under a command and control (C2, or C&C) infrastructure, a group of bots are able to form a self-propagating, self-organizing, and autonomous framework, named Botnet. Generally, to compromise a series of systems, the Botnet’s master (also called as herder or perpetrator) will remotely control bots to install worms, Trojan horses, or backdoors on them . The majority of those victims are running Microsoft

Windows operating system. The process of stealing host resources to form a Botnet is so called “scrumping” [65].

A Botnet's originator (aka "bot herder" or "bot master") can control the group remotely, usually through a means such as IRC, and usually for nefarious purposes. Individual programs manifest as IRC "bots". Often the command-and-control takes place via an IRC server or a specific channel on a public IRC network. This server is known as the command-and-control server ("C&C"). Though rare, more experienced Botnet operators program their own commanding protocols from scratch. The constituents of these protocols include a server program, client program for operation, and the program that embeds itself on the victim's machine (bot) [66]. All three of these usually communicate with each other over a network using a unique encryption scheme for stealth and protection against detection or intrusion into the Botnet network.

A bot typically runs hidden and uses a covert channel (e.g. the RFC 1459 (IRC) standard, twitter or IM) to communicate with its C&C server. Generally, the perpetrator of the Botnet has compromised a series of systems using various tools (exploits, buffer overflows, as well as others; see also RPC). Newer bots can automatically scan their environment and propagate themselves using vulnerabilities and weak passwords. Generally, the more vulnerabilities a bot can scan and propagate through, the more valuable it becomes to a Botnet controller community. The process of stealing computing resources as a result of a system being joined to a "Botnet" is sometimes referred to as "scrumping" [67].

Fortunately, Botnet attacks and the corresponding preventive measures or tracking approaches have been studied by industry and academia in last

decades. It is known that Botnets have thousands of different implementations, which can be classified into two major categories based on their topologies One typical and the most common type is Internet Relay Chat-(IRC-) based Botnets. Because of its centralized architecture, researchers have designed some feasible countermeasures to detect and destroy such Botnets. Hence, newer and more sophisticated hackers/attackers start to use Peer to Peer (P2P) technologies in Botnets. P2P Botnets are distributed and do not have a central point of failure. Compared to IRC-based Botnets, they are more difficult to detect and take down. Besides, most of its existing studies are still in the analysis phase [68].

Scholars firstly discovered Botnets due to the study on Distributed DoS (DDoS) attacks. After that, Botnet features have been disclosed using probing and Honeypots. Spammers increasingly relied on bots to generate spam messages, since bots can hide their identities. To identify and block spam, blacklists are widely used in practice. Jung and Sit found that 80% of spammers could be detected by blacklists of MIT in 2004. Besides, blacklists also impact on other hostile actions [66].

4.1.1 Classification

Botnets are emerging threats with billions of hosts worldwide infected. Bots can spread over thousands of computers at a very high speed as worms do. Unlike worms, bots in a Botnet are able to cooperate towards a common malicious purpose. For that reason, Botnets nowadays play a very important role in the Internet malware epidemic [69].

Many works try to summarize their taxonomy, using properties such as the propagation mechanism, the topology of C2 infrastructure used, the exploitation strategy, or the set of commands available to the perpetrator. So

far, Botnet's master often uses IRC protocol to control and manage the bots. For the sake of reducing Botnet's threat efficiently, scholars and researchers emphasize their studies on detecting IRC-based Botnets.

Generally speaking, the academic literature on Botnet detection is sparse. Some metrics has been presented by flow analysis on detecting Botnets. After filtering IRC session out of the traffic, flow-based methods were applied to discriminate malicious from benign IRC channels. The methods proposed by [70] combined both application and network layer analysis.

IRC activities at the application may be dealt with in layer, using information coming from the monitoring of network activities.

Some authors had introduced machine learning techniques into Botnet detection, since they led a better way to characterize Botnets. Currently, honeynets and Intrusion Detection System (IDS) are two major techniques to prevent their attacks. Honeynets can be deployed in both distributed and local context. They are capable of providing Botnet attacking information but cannot tell the details such as whether the victim has a certain worm. The IDS uses the signatures or behavior of existing Botnets for reference to detect potential attacks. Thus, to summarize the characteristics of Botnets is significant for secure networks. To the best of our knowledge, we have not found any other work about anomaly-based detection for Botnets. Before going to the discussion of Botnet attacks and preventive measures [71], we will introduce some relevant terms and classification of bots in the rest of this section.

4.1.1.1. Formation and Exploitation.

To illustrate the formation and exploitation, we take a spamming Botnet as an example [72].

A typical formation of Botnet can be described by the following steps [3], as illusterated in Figure 4.1.

(1) The perpetrator of Botnet sends out worms or viruses to infect victims' machines, whose payloads are bots.

(2) The bots on the infected hosts log into an IRC server or other communications medium, forming a Botnet.

(3) Spammer makes payment to the owner of this Botnet to gain the access right.

(4) Spammer sends commands to this Botnet to order the bots to send out spam.

(5) The infected hosts send the spam messages to various mail servers in the Internet.

Figure 4.1: Using a Botnet to send spam.

Botnets can be exploited for criminally purposes or just for fun, depending on the individuals. The next section will go into the details of various exploitations.

4.1.1.2. Botnet Lifecycle:

Figure 4.2 shows the lifecycle of a Botnet and a single bot [72].

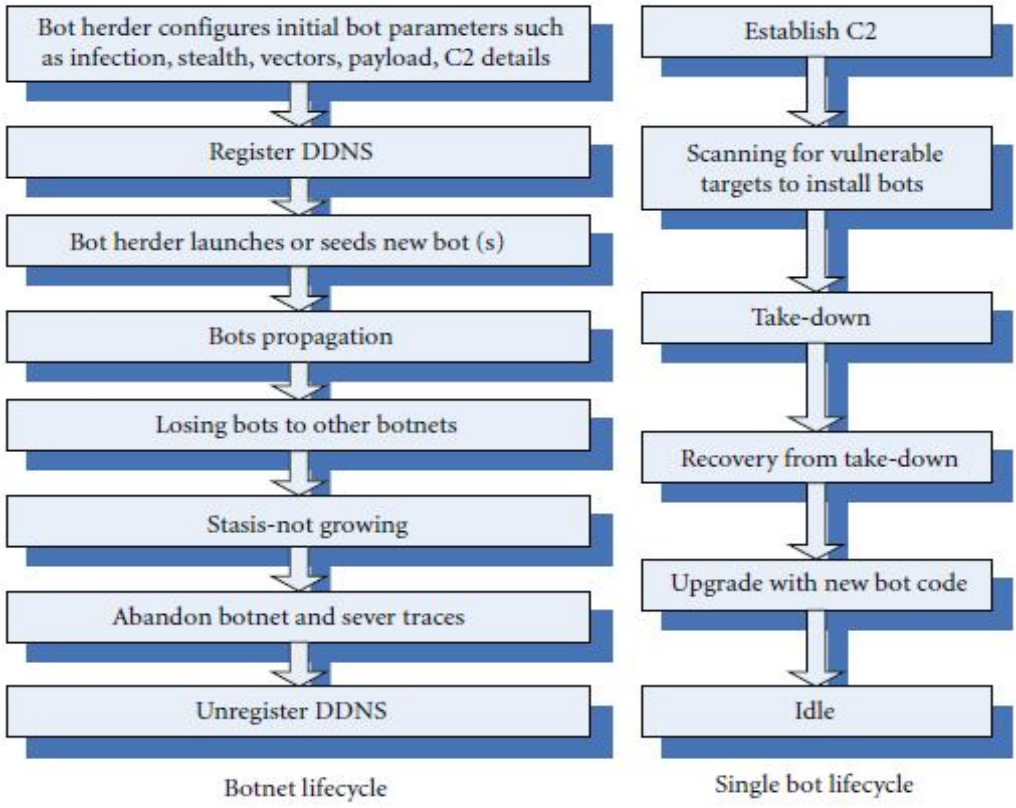


Figure 4.2: Lifecycle of a Botnet and of a single Bot [72].

4.1.1.3. IRC-Based Bot.

IRC is a protocol for text-based instant messaging among people connected with the Internet. It is based on Client/Server (C/S)model but suited for distributed environment as well [73]. Typical IRC severs are interconnected and pass messages from one to another. One can connect with hundreds of clients via multiple servers. It is so-called multiple IRC (mIRC), in which communications among clients and a server are pushed to those

who are connected to the channel. The functions of IRC-based bots include managing access lists, moving files, sharing clients, sharing channel information, and so on . Major parts of a typical IRC bot attack are showed in Figure 4.3 [74].

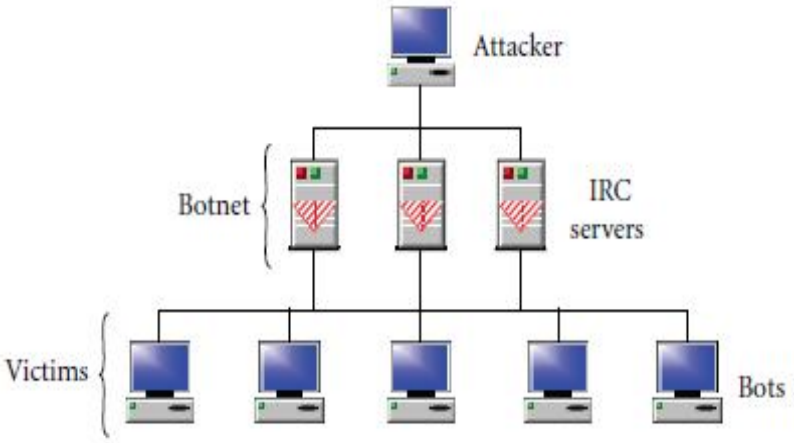


Figure 4.3: Major parts of a typical IRC Bot attack .

(i) *Bot* is typically an executable file triggered by a specific command from the IRC sever. Once a bot is installed on a victim host, it will make a copy into a configurable directory and let the malicious program to start with the operating system. Consider Windows as an instance, the bots sized no more than 15 kb are able to add into the system registry (HKEY LOCAL MACHINE\SOFTWARE\Microsoft\Windows\CurrentVerssion\Run\) [74].Generally, bots are just the payload of worms or the way to open a backdoor.

(ii) *Control channel* is a secured IRC channel set up by the attacker to manage all the bots.

(iii) *IRC Server* may be a compromised machine or even a legitimate provider for public service.

(iv) *Attacker* is the one who control the IRC bot attack. The attacker's operations have four stages [72].

(1) *The first one is the Creation Stage*, where the attacker may add malicious code or just modify an existing one out of numerous highly configurable bots over the Internet.

(2) *The second one is the Configuration Stage*, where the IRC server and channel information can be collected . As long as the bot is installed on the victim, it will automatically connect to the selected host. Then, the attacker may restrict the access and secure the channel to the bots for business or some other purpose. For example, the attacker is able to provide a list of bots for authorized users who want to further customize and use them for their own purpose.

(3) *The third one is the Infection Stage*, where bots are propagated by various direct and indirect means [75]. As the name implies, direct techniques exploit vulnerabilities of the services or operating systems and are usually associated with the use of viruses. While the vulnerable systems are compromised, they continue the infection process such that saving the time of attacker to add other victims. The most vulnerable systems are Windows 2000 and XP SP1, where the attacker can easily find unpatched or unsecured (e.g., without firewall) hosts [76]. By contrary, indirect approaches use other programs as a proxy to spread bots, that is, using distributed malware through DCC (Direct Client-to-Client) file exchange on IRC or P2P networks to exploit the vulnerabilities of target machines.

(4) *The forth one is the Control Stage*, where the attacker can send the instructions to a group of bots via IRC channel to do some malicious tasks.

4.1.1.4 P2P-Based Bot:

Few researches focus on P2P-based bots so far. It is still a challenging issue. In fact, using P2P ad hoc network to control victim hosts is not a novel technique. A worm with a P2P fashion, named Slapper [77], infected Linux system by DoS attack in 2002. It used hypothetical clients to send commands to compromised hosts and receive responses from them. Thereby, its network location could be anonymous and hardly be monitored . One year after, another P2P-based bot appeared, called Dubbed Sinit . It used public key cryptography for update authentication. Later, in 2004, Phatbot was created to send commands to other compromised hosts using a P2P system. Currently, Storm Worm may be the most wide-spread P2P bot over the Internet. Holz et al. have analyzed it using binary and network tracing . Besides, they also proposed some techniques to disrupt the communication of a P2P-based Botnet, such as eclipsing content and polluting the file. Nevertheless, the above P2P-based bots are not mature and have many weaknesses. Many P2P networks have a central server or a seed list of peers who can be contacted for adding a new peer. This process named bootstrap has a single point of failure for a P2P-based Botnet [78].
Figure 4.4 presents the C2 architecture of the hybrid P2Pbased Botnet. It has three client bots and five servant bots, who behave both as clients and servers in a traditional P2P file sharing system. The arrow represents a directed connection between bots. A group of servant bots interconnect with each other and form the backbone of the Botnet. An attacker can inject his/her commands into any hosts of this Botnet. Each host periodically connects to its neighbors for retrieving orders issued by their commander.

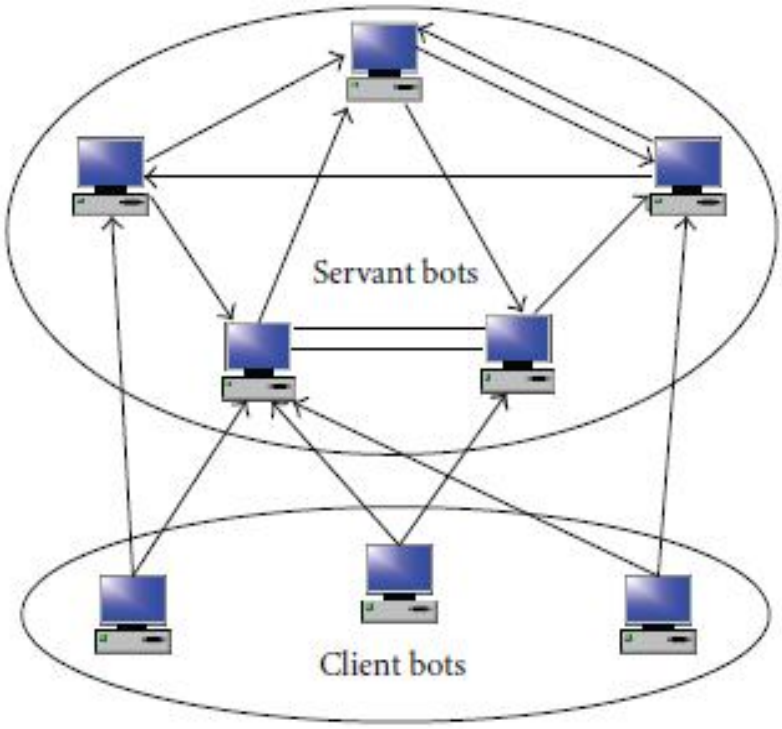


Figure 4.4: The C2 architecture of a hybrid P2P Botnetb .

As soon as a new command shows up, the host will forward this command to all nearby servant bots immediately. Such architecture combines the following features: (1) it requires no bootstrap procedure; (2) only a limited number of bots nearby the captured one can be exposed; (3) an attacker can easily manage the entire Botnet by issuing a single command [63]. Many researches proposed several countermeasures against this Botnet attack, more researches on both architecture and prevention means are still needed in the future.

4.1.1.5 Types of Bots.

Many types of bots in the network have already been discovered and studied [79]. Table 4.1 will present several widespread and well-known bots, together with their basic features. Then, some typical types will be studied in details.

Table 4.1: Types of bots.

Types	Features
Agobot Phatbot Forbot Xtrembot	They are so prevalent that over 500 variants exist in the Internet today. Agobot is the only bot that can use other control protocols besides IRC. It offers various approaches to hide bots on the compromised hosts, including NTFS Alternate Data Stream, Polymorphic Encryptor Forbot Engine and Antivirus Killer [72].
SDBot RBot UrBot UrXBot	SDBot is the basis of the other three bots and probably many more [9]. Different from Agobot, its code isunclear and only has limited functions. Even so, this group of bots is still widely used in the Internet [72].
SpyBot NetBIOS Kuang Netdevil KaZaa	There are hundreds of variants of SpyBot nowadays. Most of their C2 frameworks appear to be shared withor evolved from SDBot . But it does not provide accountability or conceal their malicious purpose in codebase [73].
mIRC-based	GT (Global Threat) bot is mIRC-based bot. It enables a mIRC chat-client based on a set of binaries (mainly

GT-Bots	DLLs) and scripts. It often hides the application window in compromised hosts to make mIRC invisible to the user [72].
DSNX Bots	The DSNX (Data Spy Network X) bot has a convenient plug-in interface for adding a new function. Albeit the default version does not meet the requirement of spreaders, plugins can help to address this problem [73].
Q8 Bots	It is designed for Unix/Linux OS with the common features of a bot, such as dynamic HTTP updating, various DDoS-attacks, execution of arbitrary commands and so forth. [80].
Kaiten	It is quite similar to Q8 Bots due to the same runtime environment and lacking of spreader as well. Kaiten has an easy remote shell, thus it is convenient to check further vulnerabilities via IRC [80].
Perl-based bots	Many variants written in Perl nowadays . They are so small that only have a few hundred lines of the bots code Thus, limited fundamental commands are available for attacks, especially for DDoS-attacks in Unix-based systems [80].

a)* **Agobot. This well-known bot is written in C/C++ with cross-platform capabilities. It is the only bot so far that utilizes a control protocol in IRC channel . Due to its standard data structures, modularity, and code documentation, Agobot is very easy for attacker to extend commands for their own purposes by simply adding new function into the C-Command-Handler or CScanner

class. Besides, it has both standard and special IRC commands for harvesting sensitive information. For example, it can request the bot to do some basic operations (accessing a file on the compromised machine by "bot.open" directive). Also, Agobot is capable of securing the system via closing NetBIOS shares, RPC-DCOM, for instance. It has various commands to control the victim host, for example, using "pctrl" to manage all the processes and using "inst" to manage autostart programs. In addition, it has the following features [81]: (1) it is IRC-based C2 framework, (2) it can launch various DoS attacks, (3) it can attack a large number of targets, (4) it offers shell encoding function and limits polymorphic obfuscations, (5) it can harvest the sensitive information via traffic sniffing (using libpcap, a packet sniffing library), key logging or searching registry entries, (6) it can evade detection of antivirus software either through patching vulnerabilities, closing back doors or disabling access to anti-virus sites (using NTFS Alternate Data Stream to hide its presence on victim host), and (7) it can detect debuggers (e.g., SoftIce and Ollydbg) and virtual machines (e.g., VMware and Virtual PC) and thus avoid disassembly . To find a new victim, Agobot just simply scans across a predefined network range. Nevertheless, it is unable to effectively distribute targets among a group of bots as a whole based on current command set [82].

b) SDBot*:* SDBot's source code is not well written in C and has no more than 2500 lines, but its command set and features are similar to Agobot . It is published under GPL. Albeit SDBot has no propagation capability and only provides some basic functions of host control**,** attackers still like this bot since its commands are easy to extend. In

addition, SDBot has its own IRC functions such as spying and cloning [77]. Spying is just recording the activities of a specified channel on a log file. Cloning means that the bot repeats to connect one channel. At present, SDBot may be the most active bot used in the wild [77]. There are plenty of auxiliary patches available on the Internet, including non malicious ones.

SDBot's is essentially a compact IRC implementation . To contact the IRC server, it first sends identity information, for example, USER and NICK . As long as it gets an admission message (PING) from the server, the bot will acknowledge this connection with a PONG response [83].While the bot receives the success code (001 or 005) for connection, it can request a hostname by USERHOST and join the channel by JOIN message. Once it receives a response code 302, this bot has successfully participated in the IRC channel and the master can control it via some IRC commands (e.g., NOTICE, PRIVMSG, or TOPIC) . With the help of many powerful scanning tools, SDBot can easily find the next victim For instance, using NetBIOS scanner, it can randomly choose a target located in any predefined IP range. Since the SDBot is able to send ICMP and UDP packets, it is always used for simple flooding attacks. Moreover, a large number of variants capable of DDoS attack are available in the wild.

a) **SpyBot. SpyBot:** is written in C with no more than 3,000 lines, and has pretty much variants nowadays as well [84]. As a matter of fact, SpyBot is enhanced version of SDBot. Besides the essential command language implementation, it also involves the scanning capability, host control function, and the modules of DDoS attack

and flooding attack (e.g., TCP SYN, ICMP, and UDP). SpyBot's host control capabilities are quite similar to Agobot's in remote command execution, process/system manipulation, key logging, and local file manipulation [77]. Nevertheless, SpyBot still does not have the capability breadth and modularity of Agobot.

d). GT Bot. GT (Global Threat) Bot, as known as Aristotles, is supposed to stand for all mIRC-based bots which have numerous variants and are widely used forWindows [85]. Besides some general capabilities such as IRC host control, DoS attacks, port scanning, and NetBIOS/RPC exploiting, GT Bot also provides a limited set of binaries and scripts of mIRC . One important binary is *HideWindow* program used to keep the mIRC instance invisible from the user . Another function is recording the response to each command received by remote hosts . Some other binaries mainly extend the functions of mIRC via DDL (Dynamic Link Library) . These scripts often store in files with "*.mrc*" extension or in "*mirc.ini*". Although the binaries are almost all named as "*mIRC.exe"*, they may have different capabilities due to distinct configuration files. Compared to the above instances, GT Bot only provides limited commands for host control, just capable of getting local system information and running or deleting local files [72].

4.1.2 Botnets Organization and Formation

Botnet servers will often liaise with other Botnet servers, such that a group may contain 20 or more individual cracked high-speed connected machines as servers, linked together for purposes of greater redundancy.

Actual Botnet communities usually consist of one or several controllers that rarely have highly-developed command hierarchies between themselves; they rely on individual friend-to-friend relationships.

The architecture of Botnets has evolved over time, and not all Botnets exhibit the same topology for command and control. Depending upon the topology implemented by the Botnet, it may make it more resilient to shutdown, enumeration, Command and control location discovery. However, some of these topologies limit the saleability and rental potential of the Botnet to other third-party operators [88]. Typical Botnet topologies are:

- Star * Multi-server * Hierarchical
- Random

To thwart detection, some Botnets were scaling back in size. As of 2006, the average size of a network was estimated at 20,000 computers, although

The example in figure 4.5 illustrates how a Botnet is created and used to send email spam.

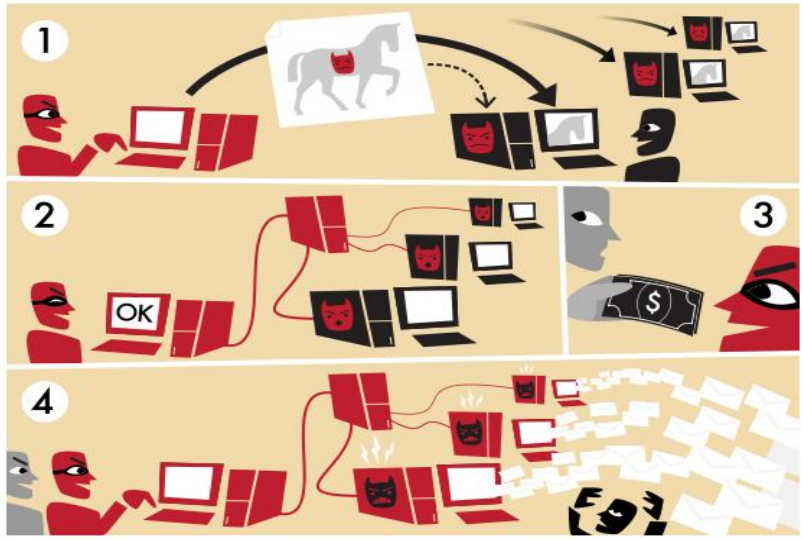


Figure 4.5: Botnet via spam mail

A Botnet operator sends out viruses or worms, infecting ordinary users' computers, whose payload is a malicious application—the *bot*.
The *bot* on the infected PC logs into a particular C&C server (often an IRC server, but, in some cases a web server).

A spammer purchases access to the Botnet from the operator. The spammer sends instructions via the IRC server to the infected PCs, causing them to send out spam messages to mail servers.

Botnets are exploited for various purposes, including denial-of-service attacks, creation or misuse of SMTP mail relays for spam, click fraud, spamdexing and the theft of application serial numbers, login IDs, and financial information such as credit card numbers [87].

The Botnet controller community features a constant and continuous struggle over who has the most bots, the highest overall bandwidth, and the most "high-quality" infected machines, like university, corporate, and even government machines.

4.1.3 Botnet Attacks:

Botnets can serve both legitimate and illegitimate purposes [86]. One legitimate purpose is to support the operations of IRC channels using administrative privileges on specific individuals. Nevertheless, such goals do not meet the vast number of bots that we have seen. Based on the wealth of data logged in Honeypots, the possibilities to use Botnets for criminally motivated or for destructive goals can be categorized as follows.

4.1.3.1. DDoS Attacks:

Botnets are often used for DDoS attacks, which can disable the network services of victim system by consuming its bandwidth. For instance,

a perpetrator may order the Botnet to connect a victim's IRC channel at first, and then this target can be flooded by thousands of service requests from the Botnet [88]. In this kind of DDoS attack, the victim IRC network is taken down. Evidence reveals that most commonly implemented by Botnets are TCP SYN and UDP flooding attacks.

General countermeasure against DDoS attacks requires:

(1) controlling a large number of compromised machines;

(2) disabling the remote control mechanism . However, more efficient ways are still needed to avoid this kind of attack. Freiling et al. [119] have presented an approach to prevent DDoS attack via exploring the hiding bots in Honeypots.

4.1.3.2. Spamming and Spreading Malware:

About 70% to 90% of the world's spam is caused by Botnets nowadays, which has most experienced in the Internet security industry concerned. Study report indicates that, once the SOCKS v4/v5 proxy (TCP/IP RFC 1928) on compromised hosts is opened by some bots, those machines may be used for nefarious tasks, for example, spamming. Besides, some bots are able to gather email addresses by some particular functions. Therefore, attackers can use such a Botnet to send massive amounts of spam [73]. Researchers in many fields have proposed a distributed content independent spam classification system, called Trinity, against spamming from Botnets. The designer assumes that the spamming bots will send a mass of e-mails within a short time. Hence, any letter from such address can be a spam. It is a little bit unexpected that we do not know the effectiveness of Trinity since it is still under experiment. In order to discover the aggregate behaviors of spamming Botnet and benefit its detection in the future. Spam

signature has been deployed as a generation framework named AutoRE. They also found several characteristics of spamming Botnet:

(1) spammer often appends some random and legitimate URLs into the letter to evade detection ;

(2) Botnet IP addresses are usually distributed over many ASes (Autonomous Systems), with only a few participating machines in each AS on average [63];

(3) despite that the contents of spam are different, their recipients' addresses may be similar [65]. How to use these features to capture the Botnets and avoid spamming is worth to research in the future.

Similarly, Botnets can be used to spread malware too. For instance, a Botnet can launch Witty worm to attack ICQ protocol since the victims' system may have not activated Internet Security Systems (ISS) services [85].

4.1.3.3 Information Leakage:

Because some bots may sniff not only the traffic passing by the compromised machines but also the command data within the victims, perpetrators can retrieve sensitive information like usernames and passwords from Botnets easily. Evidences indicate that, Botnets are becoming more sophisticated at quickly scanning in the host for significant corporate and financial data. Since the bots rarely affect the performance of the running infected systems, they are often out of the surveillance area and hard to be caught. Keylogging is the very solution to the inner attack . Such kind of bots listens for keyboard activities and then reports to its master the useful information after filtering the meaningless inputs. This enables the attacker to steal thousands of private information and credential data [86].

4.1.3.4. Click Fraud*:*

With the help of Botnet, perpetrators are able to install advertisement add-ons and browser helper objects (BHOs) for business purpose. Just like Google's AdSense program, for the sake of obtaining higher click-through rate (CTR), perpetrators may use Botnets to periodically click on specific hyperlinks and thus promote the CTR artificially [87]. This is also effective to online polls or games. Because each victim's host owns a unique IP address scattered across the globe, every single click will be regarded as a valid action from a legitimate person.

4.1.3.5. Identity Fraud:

Identity Fraud, also called as Identity Theft, is a fast growing crime on the Internet. Phishing mail is a typical case. It usually includes legitimate-like URLs and asks the receiver to submit personal or confidential information. Such mails can be generated and sent by Botnets through spamming mechanisms [88]. In a further step, Botnets also can set up several fake websites pretending to be an official business sites to harvest victims' information. Once a fake site is closed by its owner, another one can pop up, until you shut down the computer.

4.1.1 Detection and tracing of Botnets:

By now, several different approaches of identifying and tracing back Botnets have been proposed or attempted. First and the most generally, the use of Honeypots, where a subnet pretends to be compromised by a Trojan, but actually observing the behavior of attackers, enables the controlling hosts

to be identified [82]. In a relevant case, it has introduced a feasible way to detect certain types of DDoS attacks lunched by the Botnet. To begin with, use honeypot and active responders to collect bot binaries. Then, pretend to join the Botnet as a compromised machine by running bots on the honeypot and allowing them to access the IRC server. At the end, the Botnet is infiltrated by a "silent drone" for information collecting, which may be useful method is using the information form insiders to track an IRC-based Botnet. The third but not the least prevalent approach to detect Botnets is probing DNS caches on the network to resolve the IP addresses of the destination servers [88].

4.1.4.1. Honeypot and Honeynet:

Honeypots are well-known by their strong ability to detect security threats, collect malwares, and to understand the behaviors and motivations of perpetrators. Honeynet, for monitoring a large-scale diverse network, consists of more than one honeypot on a network.

Most of researchers focus on Linux-based honeynet, due to the obvious reason that, compared to any other platform, more freely honeynet tools are available on Linux [89]. As a result, only few tools support the honeypots deployment onWindows and intruders start to proactively dismantle the honeypot.

Some scholars aim at the design of a reactive firewall or relatedmeans to preventmultiple compromises of honeypots. While a compromised port is detected by such a firewall, the inbound attacks on it can be blocked. This operation should be carried on covertly to avoid raising suspicions of the attacker. Evidence shows that operating less covertly is needed on protection

of honeypots against multiple compromises by worms, since worms are used to detect its presence . Because many intruders download toolkits in a victim immediate aftermath, corresponding traffic should be blocked only selectively. Such toolkits are significant evidences for future analysis. Hence, to some extent, attackers' access to honeypots could not be prevented very well [90].

As honeypots have become more and more popular in monitoring and defense systems, intruders begin to seek a way to avoid honeypot traps. There are some feasible techniques to detect honeypots. For instance, to detect VMware or other emulated virtual machines , or, to detect the responses of program's faulty in honeypot. Researchers inthis particular point *have* successfully identified honeypots using intelligent probing according to public report statistics. In addition, a technique is presented a commercial spamming tool capable of anti-honeypot function, called "Send-Safe's Honeypot Hunter." By checking the reply formremote proxy, spammer is able to detect honeypot open proxies . However, this tool cannot effectively detect others except open proxy honeypot. Recently, a mechanism has been proposed that uses another methodology for honeypot detection based on independent software and hardware. In their paper, they also have introduced an approach to effectively locate and remove infected honeypots using a P2P structured Botnet [91]. All of the above evidences indicate that, future research is needed in case that a Botnet becomes invisible to honeypot.

4.1.4.2 IRC-based Detection:

IRC-based Botnet is wildly studied and therefore several characteristics have been discovered for detection so far. One of the easy

ways to detect this kind of Botnets is to sniff traffic on common IRC ports (TCP port 6667), and then check whether the payloads march the strings in the knowledge database [92]. Nevertheless, Botnets can use random ports to communicate. Therefore, another approach looking for behavioral haracteristics of bots comes up. IRC-based bots are often idle and only responded upon receiving a specific instruction. Thus, the connections with such features can be marked as potential enemies. Nevertheless, it still has a high false positive rate in the result.

There are also other methodologies existing for IRCbased Botnet detection [77]. some approaches has been proposed based on the source code analysis introducing a modified IRC client called IRC tracker, which was able to connect the IRC sever and reply the queries automatically. Given a template and relevant fingerprint, the IRC tracker could instantiate a new IRC session to the IRC server. In case the bot master could find the real identity of the tracker, it appeared as a powerful and responsive bot on the Internet and run every malicious command, including the responses to the attacker [81]. We will introduce some detection methods against IRC-based Botnets below.

a) Detection Based on Traffic Analysis:. Signature technology is often used in anomaly detection. The basic idea is to extract feature information on the packets from the traffic and march the patterns registered in the knowledge base of existing bots. Apparently, it is easy to carry on by simply comparing every byte in the packet, but it also goes with several drawbacks . Firstly, it is unable to identify the undefined bots. Second, it should always update the knowledge base with new signatures, which enhances the management cost and reduces the performance [83]. Third, new bots may launch attacks before the knowledge base are patched .

Based on the features of IRC, some other techniques to detect Botnets come up. Basically, two kinds of actions are involved in a normal IRC communication. One is interactive commands and another is messages exchanging. If we can identify the IRC operation with a specified program, it is possible to detect a Botnet attack [81]. For instance, if the private information is copied to other places by some IRC commands, we claim that the system is under an attack since a normal chatting behavior will never do that. However, the shortcomings also exist. On the one hand, IRC port number may be changed by attackers. On the other hand, the traffic may be encrypted or be concealed by network noises [86]. Any situation will make the bots invisible. Many researchers observed the real traffic on IRC communication ports ranging from 6666 to 6669. They found some IRC clients repeated sending login information while the server refused their connections. Based on the experiment result, they claimed that bots would repeat these actions at certain intervals after refused by the IRC server, and those time intervals are different . However, they did not consider a real IRC-based Botnet attack into their experiment. It is a possible future work to extend their achievements.

Other researchs proposes a different method for Botnet detection. Their approach can efficiently and automatically identify spam or bots. The main idea is to extract the shape of the Email (lines and the character count of each line) by applying a Gaussian kernel density estimator [93]. Emails with similar shape are suspected. However, authors did not show the way to detect Botnet by using this method. It may be another future work worth to study.

b) Detection Based on Anomaly Activities: an algorithm for anomaly-based Botnet detection. It combined IRC mesh features with TCP-

based anomaly detection module. It first observed and recorded a large number of TCP packets with respect to IRC hosts. Based on the ratio computed by the total amount of TCP control packets (e.g., SYN, SYNACK, FIN, and RESETS) over total number of TCP packets, it is able to detect some anomaly activities . They called this ratio as the TCP work weight and claimed that high value implied a potential attack by a scanner or worm. However, this mechanism may not work if the IRC commands are not valid [94].

4.1.4.3 DNS Tracking:

Since bots usually send DNS queries in order to access the C2 servers, if we can intercept their domain names, the Botnet traffic is able to be captured by blacklisting the domain names Actually, it also provides an important secondary avenue to take down Botnets by disabling their propagation capability. The features of Botnet DNS has been discussed. According to Botnet analysis, Botnets' DNS queries can be easily distinguished from legitimate ones [95].

First of all, only bots will send DNS queries to the domain of C2 servers, a legitimate one never do this. Secondly, Botnet's members act and migrate together simultaneously, as well as their DNS queries. Whereas the legitimate one occurs continuously, varying from Botnet. Third, legitimate hosts will not use DDNS very often while Botnet usually use DDNS for C2 servers.

Based on the above features, they developed an algorithm to identify Botnet DNS queries. The main idea is to compute the similarity for group activities and then distinguish the Botnet from them based on the similarity value. The similarity value is defined as 0.5 (C/A+C/B), where A and B

stand for the sizes of two requested IP lists which have some common IP addresses and the same domain name, and C stands for the size of duplicated IP addresses . If the value approximated zero, such common domain will be suspected.

There are also some other approaches. Dagon [71] presented a method of examining the query rates of DDNS domain. Abnormally high rates or temporally concentrated were suspected, since the attackers changed their C2 servers quite often. They utilized both Mahalanobis distance and Chebyshev's inequality to quantify how anomalous the rate is .it had been found that when C2 servers had been taken down, DDNS would often response name error. Hosts who repeatedly did such queries could be infected and thus to be suspected. The evaluation of the above two methods through experiments on the real world. They claimed that, the first approach was not as effective since it misclassified some C2 server domains with short TTL, while the second method was comparatively effective due to the fact that the suspicious name came from independent individuals [77].

A Botnet detection system called RB-Seeker (Redirection Botnet Seeker) had been proposed. It is able to automatically detect Botnets in any structure. RB-Seeker first gathers information about bots redirection activities (e.g., temporal and spatial features) from two subsystems. Then it utilizes the statistical methodology and DNS query probing technique to distinguish the malicious domain from legitimate ones. Experiment results show that RB-Seeker is an efficient tool to detect both "aggressive" and "stealthy" Botnets.

4.1.5. Preventive Measures

If a machine receives a denial-of-service attack from a Botnet, few choices exist. Given the general geographic dispersal of Botnets, it becomes difficult to identify a pattern of offending machines, and the sheer volume of IP addresses does not lend itself to the filtering of individual cases. Passive OS fingerprinting can identify attacks originating from a Botnet: network administrators can configure newer firewall equipment to take action on a Botnet attack by using information obtained from passive OS fingerprinting. The most serious preventive measures utilize rate-based intrusion prevention systems implemented with specialized hardware [81].

Some Botnets use free DNS hosting services such as DynDns.org, No-IP.com, and Afraid.org to point a subdomain towards an IRC server that will harbor the bots. While these free DNS services do not themselves host attacks, they provide reference points (often hard-coded into the Botnet executable). Removing such services can cripple an entire Botnet. Recently, these companies have undertaken efforts to purge their domains of these subdomains. The Botnet community refers to such efforts as "nullrouting", because the DNS hosting services usually re-direct the offending subdomains to an inaccessible IP address [93].

The Botnet server structure mentioned above has inherent vulnerabilities and problems. For example, if one was to find one server with one Botnet channel, often all other servers, as well as other bots themselves, will be revealed. If a Botnet server structure lacks redundancy, the disconnection of one server will cause the entire Botnet to collapse, at least until the controller(s) decides on a new hosting space. However, more

recent IRC server software includes features to mask other connected servers and bots, so that a discovery of one channel will not lead to disruption of the Botnet.

Several security companies such as Afferent Security labs, Symantec, Trend Micro, FireEye, Simplicita and Damballa have announced offerings to stop Botnets. While some, like NortonAntiBot, are aimed at consumers, most are aimed to protect enterprises and/or ISPs [94]. The host-based techniques use heuristics to try to identify bot behavior that has bypassed conventional anti-virus software. Network-based approaches tend to use the techniques described above; shutting down C&C servers, nullrouting DNS entries, or completely shutting down IRC servers.

Newer Botnets are almost entirely P2P, with command-and-control embedded into the Botnet itself. By being dynamically updateable and variable they can evade having any single point of failure. Commanders can be identified solely through secure keys and all data except the binary itself can be encrypted. For example a spyware program may encrypt all suspected passwords with a public key hard coded or distributed into the bot software. Only with the private key, which only the commander has, can the data that the bot has captured be read [95].

Newer Botnets have even been capable of detecting and reacting to attempts to figure out how they work. A large Botnet that can detect that its being studied can even DDoS those studying it off the internet.

There is an effort by researchers at Sandia National Laboratories to analyze the behavior of these Botnets by simultaneously running one million Linux kernels as virtual machines on a 4,480-node Dell high-performance computer cluster.

It takes only a couple of hours for conventional worms to circle the globe since its release from a single host. If worms using Botnet appear from multiple hosts simultaneously, they are able to infect the majority of vulnerable hosts worldwide in minutes [87]. Some Botnets have been discussed in previous sections. Nevertheless, there are still plenty of them that are unknown to us.We also discuss a topic of how to minimize the risk caused by Botnets in the future in this section.

4.1.5.1 Countermeasures on Botnet Attacks:

Unfortunately, few solutions have been in existence for a host to against a Botnet DoS attack so far. Albeit it is hard to find the patterns of malicious hosts, network administrators can still identify Botnet attacks based on passive operating system fingerprinting extracted from the latest firewall equipment. The lifecycle of Botnets tells us that bots often utilize free DNS hosting services to redirect a sub-domain to an inaccessible IP address. Thus, removing those services may take down such a Botnet . At present, many security companies focus on offerings to stop Botnets . Some of them protect consumers, whereas most others are designed for ISPs or enterprises. The individual products try to identify bot behavior by anti-virus software [96]. The enterprise products have no better solutions than nullrouting DNS entries or shutting down the IRC and othermain servers after a Botnet attack identified.

4.1.5.2 Countermeasures for Public:

Personal or corporation security inevitably depends on the communication partners. Building a good relationship with those partners is

essential. Firstly, one should continuously request the service supplier for security packages, such as firewall, anti-virus tool-kit, intrusion detection utility, and so forth.Once something goes wrong, there should be a corresponding contact number to call. Secondly, one should also pay much attention on network traffic and report it to ISP if there is a DDoS attack. ISP can help blocking those malicious IP addresses. Thirdly, it is better to establish accountability on its system, together with a law enforcement authority. More specifically, scholars and industries have proposed some strategies for both home users and system administrators, to prevent, detect and respond Botnet attacks [97]. Here we summarize their suggestions.

a) Home Users. To prevent attacks from a Botnet, home users can follow the rules described in Table 4.2. They are classified into three categories: (1) Personal (2) Routine, (3) Optional Operations.

Table 4.2: Rules of prevention by home users.

Type	Strategies
Personal Habits	Attention while downloading Avoid to install useless things Read carefully before you click
Routine	Use anti-virus/trojan utilities Update system frequently Shutdown PC when you leave
Optional Operations	Back-up all systems regularly Keep all software up-to-date Deploy personal firewall

As personal habits, people should pay attention when downloading, especially for those programs coming from unscrupulous sites. Besides, try to avoid installing useless things on personal computer, which will minimize the possibility of bots infection. If necessary, read the License Agreement and the notes carefully before click the button on the web site. As a routine, use antivirus software and anti-trojan utilities while system is on [98].

Scan and update system regularly, especially for Windows. When leaving the PC, shutdown the system or it may be remotely controlled by hackers. As the optional operations, home users are recommended to backup system regularly, to keep all software up-to-date and to deploy personal firewall by allmeans. By doing so, home PCs are shielded fromunauthorized accesses, and thus bots cannot compromise them.

To detect an abnormal behavior, taking Windows operating system as an instance, a home user can check the IRC port range from 6000 to 7000 (typically 6667) by command "C:\Windows\netstat-an" . The result can reveal the connection of current IRC client. However, bots may use some other TCP ports [96]. If unusual behavior occurs on a home PC, such as slow network response, unknown ports being used, and something like that, there is possibly a bot attack. Also, home users can use anti-virus software or online services to detect attacks. Once the computer has been compromised, there are strategies to recover it. The following procedure depicted in figure 5.6 is a good example for home users.

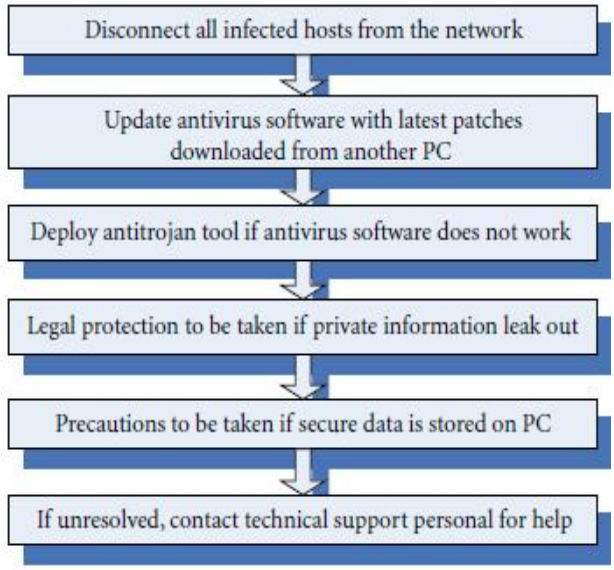


Figure 4.6: Home users' response to Botnet attacks.

b) System Administrator. Similarly, there are corresponding rules for system dministrators to prevent, detect, and respond Botnet attacks [98]. For a prevention method, administrators should follow vendor guidelines for updating the system and applications. Also, keep informed of latest vulnerabilities and use access control and log files to achieve accountability. As illustrated in Table 5.3, the following measures can help the system administrator to minimize the possibilities of Botnet attacking. Once an attack is detected, a system administrator should isolate those compromised hosts and notify the home users. Then preserve the data on those infected hosts including the log files. Besides, identify the number of victims via sniffer tools. Finally, report the infection to security consultant.

Table 5.3 Rules of detection by system administrators

Rules	Notes
Monitor all log filess regularly	Analyze & Report the internet traffic for anomalie
Use network packet sniffer	Identify the malicious traffic in intranet
Isolate the malicious subnet	Verify IRC activity on host
Scan individual machine	They may contain malware

4.2 Botnet Prevention in MANET based on PKI using fuzzy function

In this section, a Security algorithm applied to MANETs is presented. This algorithm may be viewed as a two stages: first a fuzzy model to decide the key length for the current session. Then the key distribution between nodes in MANET both stages are illustrated in the rest of this section. By doing so the Botnet threat is minimized in an obvious manner as the experimental results will show, because each new node will be subjected to the PKI that will prevent any misbehaviouring node from entering or even registering to the MANET's log table. Hence any rejected node will not be capable of receiving or sending data to any other node in the MANET.

4.2.1 Fuzzy model for Key Size Determination Function

The security offered by the algorithm is based on the difficulty of discovering the secret key through a brute force attack. Mobile Status (MS) Security Level is the correlative factor being analyzed with three considerations:

(1) The longer the password, harder to withstand a severe attack of brute force. In this research the key lengths from 16 to 512 are assumed

(2) The quickest way to change passwords, more secure the mobile host. It is more difficult to decipher the key to a shorter time. A mobile host to change the secret key is often safer than a mobile host using a constant secret key.

(3) The neighbor hosts the mobile host has, the more potential attacker. I.e. the possibility of attack is greater. There are many other factors affecting the safety of mobile hosts, such as bandwidth. The security level of mobile hosts is a function with multiple variables and affected more than one condition.

Here a fuzzy logic system is defined . Inputs of the fuzzy logic system are the frequency of changing keys *(f)* and the number of neighbor hosts *(n)*. Output of the fuzzy logic system is the Security-Level of M*S*. It is assumed that the three factors are independent with each other. The relationship of them is as follows:

$$S \propto l \bullet f \bullet \frac{1}{n}.$$ Formula 1

It means that the Security-Level of MH is in direct proportion to the length of the key and the frequency of changing keys, in inverse proportion to the number of neighbor hosts. The *S* value is updated by the fuzzy logic system. When the key length is short, the Security-Level of MH should be low; otherwise the Security-Level of MS should be high.

- The input fuzzy variable ―the number of neighbor hosts‖ has three fuzzy sets—few, normal and many.
  The membership function of *n* is illustrated in Figure 4.7.

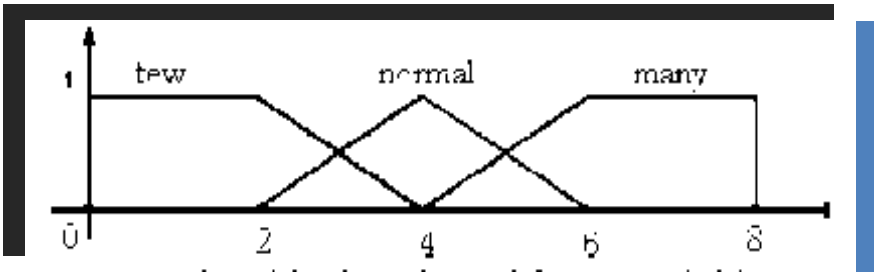


Figure 4.7: Membership function of fuzzy variable *n*.

- The input fuzzy variable ―the frequency of changing keys‖ has two fuzzy sets—slow and fast. The membership functions of *f* is showed in figure 4.8

$$f = \begin{cases} slow & \text{the secret key is constant} \\ fast & \text{the secret key is variable.} \end{cases}$$

Figure 4.8 the formula for the parameter f

- The output fuzzy variable ―the Security-Level of MS‖ has five fuzzy sets -lowest, low, normal, high and highest. It should be noted that modifying the membership functions will change the sensitivity of the fuzzy logic system's output to its inputs. Also increasing the number of fuzzy sets of the variables will provide better sensitivity control but also increases computational complexity of the system. Table 4.3 shows the rules used in the fuzzy logic system.

Table 4.3: the fuzzy system rules

Input		Output
F	N	S
Slow	Few	Low
Slow	Normal	Lowest
Slow	Many	Lowest
Fast	Few	Normal
Fast	Normal	Low
Fast	Many	Low
Slow	Few	High
Slow	Normal	Normal
Slow	Many	Low
Fast	Few	Highest
Fast	Normal	High
Fast	Many	High

The output of that system determines the number of bits used and the security level required for the current situation will follow the following fuzzy rules:

S is lowest: the number of bits is 16;

S is low: the number of bits is 32;

S is normal: the number of bits is 64;

S is high: the number of bits is 128;

S is highest: the number of bits is 256 or 512.

4.2.2 key distribution:

Once the fuzzy function has decided the length of the session key based on its criteria the problem of key creation and distribution arises. The nature of NANET poses great challenges due to the lake of infrastructure and control over the network. To overcome such problems the use of PK scheme is used to distribute the key under the assumption that one node (let us say the first node that originates the network) is responsible for the creation of session keys. If that node is going to leave the network it must transfer the process of key creation to another trusted node in the network.

1- Each node sends a message (Session Key Request SKR) encrypted with its private key (that message contains a key request and a timer) to the key creator node which owns a table that contains the public key for each node in the network. Figure 4.9 (a) where the direction of the arrow's head denotes the private key used encryption is the originating node.

2- The key creator node simply decrypts the message and retrieves the request and the timer with one of the following scenarios occurs:

   a. The timer was expired or the message is unreadable the message is neglected.

b. The timer is valid and the decryption of the message using the corresponding Public Key gives a readable request. The key creator node sends a message to that node containing the current session key. That message is encrypted two times first using the key creator's Private key(for authentication) then using the destination's public key Figure 4.9 (b). Where the direction of the arrow's head denotes the private key used encryption is the trusted node then with the destination node's Public Key.

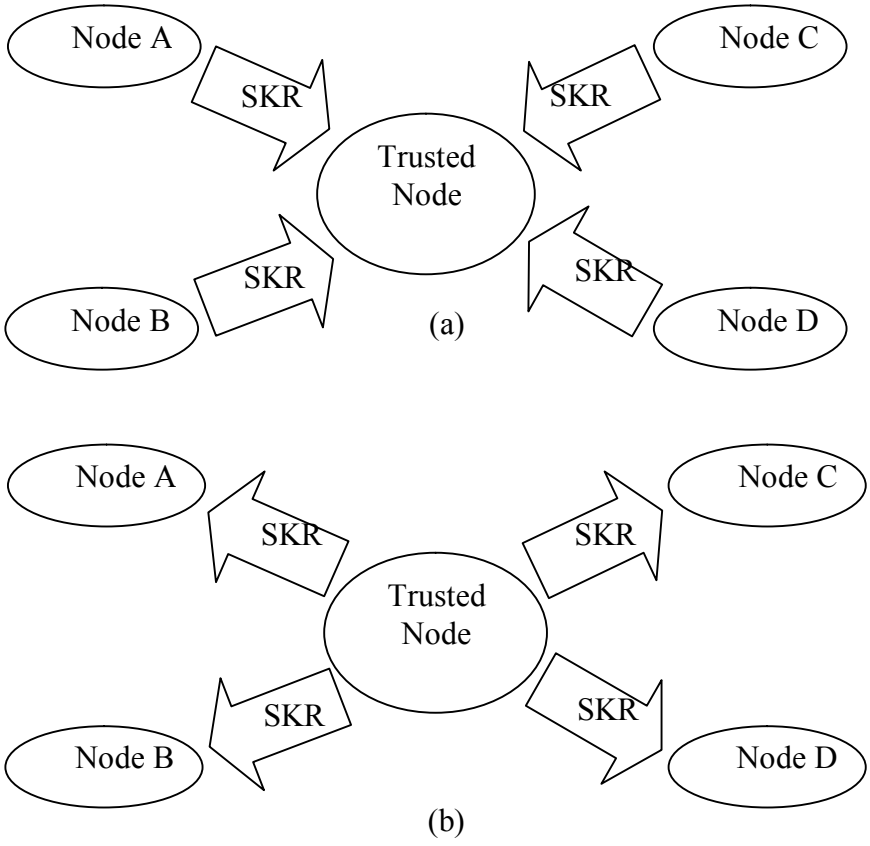


Figure 4.9 key distribution : (a) SKR (b)SKR reply

3- Any time the fuzzy model reports that the network condition changes; the key creator node sends a jamming message for every node currently in the network asking them to send a key request message.

4- Any authenticated node (including the Trusted node) on the network knowing the current session key can send messages either to every node or to a single node on the network, simply by encrypting the message using the current session key. .

Chapter 5: AIS Model for Botnets Manipulation in MANETs Using Fuzzy Function

In this chapter the Artificial Immune System (AIS) is shown to carry the whole security processes in MANETs. The chapter is provided into two subsections; the first sis section 5.1 which provides the necessary review and knowledge required to understand the interesting environment of AIS. Section 5.2 shows the proposed model for security in MANETs using AIS applied to face the dangerous of the Botnets for detection if any node becomes misbehavioring.

5.1 AIS overview

The biological immune system (IS) is highly complicated and appears to be precisely tuned to the problem of detecting and eliminating infections. We believe that the IS provides a compelling example of a massively-parallel adaptive information-processing system, one which we can study for the purpose of designing better artificial systems [99]. The IS is compelling because it exhibits many properties that we would like to incorporate into artificial systems: It is diverse, distributed, error tolerant, dynamic, self-monitoring (or self-aware) and adaptable. These properties give the IS certain key characteristics that most artificial systems today lack: robustness, adaptivity and autonomity.

Robustness is a consequence of the fact that the IS is diverse, distributed, dynamic and error tolerant. *Diversity* improves robustness on both a population and individual level, for example, different people are vulnerable to different infections [100]. The IS is *distributed* in a robust fashion: its

many components interact locally to provide global protection, so there is no central control and hence no single point of failure. The IS is *dynamic* that individual components are continually created, destroyed, and circulated throughout the body, which increases the temporal and spatial diversity of the IS. Finally, the IS is robust to errors (*error tolerant*) because the effect of any single IS action is small, so a few mistakes in classification and response are not catastrophic. The IS is *adaptable* in that it can learn to recognize and respond to new infections and retain a memory of those infections to facilitate future responses. This adaptivity is made possible by the *dynamic* functioning of the IS, which enables the IS to discard components which are useless or dangerous and to improve on existing components [101].

The IS is *autonomous* in that there is no outside control required, and the IS is an integrated part of the body, and hence the same mechanisms that monitor and protect the rest of the body also monitor and protect the IS. Furthermore, the distributed, decentralized nature of the IS contributes to its autonomous nature: not only is there no outside control, but there is no way of imposing outside control or even inside, centralized control. These properties of robustness, adaptability, turnover of components, and autonomy are closely related to the design principles of complex adaptive systems articulated by Holland [102]. Furthermore, the immune system appears to reflect many aspects of a less well-articulated design illustrated by Holland's genetic algorithms, classifier systems, and Echo. Common features in those systems include: fine-grained representations and actions, emergence of coordinated behavior, competition among components, random variation, evolution, and close coupling with a perpetually novel environment.

Representations and actions in the immune system are fine-grained (short protein fragments, called peptides, are the basic unit of representation). Coherent coordinated behavior arises (or *emerges*) from the interactions of literally trillions of cells and molecules. Each individual action of the immune system (forming a chemical bond, secreting molecules from a cell, killing a single cell, etc.) is also fine-grained. Another feature of the Holland design is the notion of competition for survival among the basic units of an adaptive system [101]. This is seen in the immune system when individual immune cells compete with one another to bind foreign antigen. Immune receptors are created randomly through genetic recombinations and mutations. Mutations take place when gene fragments are joined into a single gene (junctional diversity) and during affinity maturation (somatic hypermutation). Evolutionary processes play a central role in the Holland aesthetic.

The immune system illustrates the use of evolution as an engine of innovation in its affinity maturation of B-cells in response to foreign antigen, which quite closely resembles a genetic algorithm without crossover. Finally, the notion of an adaptive system being closely coupled with its environment, responding to perpetually novel stimulii in a dynamic and flexible way, is a basic tenet of Holland's view of adaptive systems. This view is perhaps better illustrated by classifier systems and Echo than by conventional genetic algorithms [103].

The idea is to describe a system called (AIS_1) which incorporates these properties. To preserve generality, AIS is described independently of any particular problem domain. However, to ground these concepts, we *situate* AIS in a networked environment as a computer security system called LISYS. Many researches shows that it is fruitless to design intelligent

systems in complete isolation from the environments in which they exist. The hope is that situating an intelligent artifact will simplify it, because it can use its environment to reduce computations, and it will be less likely to include unnecessary features or mechanisms [104].

Computer security is an important and natural application domain for adaptive systems. Computer systems are dynamic, with continually changing patterns of behavior; programs are added and removed, new users are introduced, configurations change. These and other changes allow intruders to find novel means to gain improper access to computers. Traditional computer security mechanisms are largely static and so cannot easily cope with dynamic environments. It is known that an adaptive system is needed to track both changes in the environment and the way in which intruders exploit systems. A computer security system should protect a machine or set of machines from intruders and foreign code, which is similar in functionality to the immune system protecting the body from invasion by inimical microbes. Because of these similarities, we have designed and implemented LISYS, an intrusion detection system that monitors network traffic. LISYS demonstrates the utility of AIS when applied to a specific problem domain [101].

5.1.1 The immune system

The IS consists of a multitude of cells and molecules which interact in a variety of ways to detect and eliminate infectious agents (pathogens). These interactions are localized because they depend upon chemical bonding—surfaces of immune system cells are covered with receptors, some of which chemically bind to pathogens, and some of which bind to other immune system cells or molecules to enable the complex system of signaling

that mediates the immune response [74]. Most IS cells circulate around the body via the blood and lymph systems, forming a dynamic system of distributed detection and response, where there is no centralized control, and little, if any or hierarchical organization. Detection and elimination of pathogens are consequences of trillions of cells interacting through simple, localized rules. A consequence of this is that the IS is very robust to failure of individual components and attacks on the IS itself.

The problem of detecting pathogens is often described as that of distinguishing "self" from "nonself" (which are elements of the body, and pathogens, respectively) [105]. However,many pathogens are not harmful, and an immune response to eliminate them may damage the body. In these cases it would be healthier not to respond, so it would be more accurate to say that the problem faced by the IS is that of distinguishing between *harmful* nonself and everything else. the viewpoint has been adopted that "nonself" is synonymous with any pathogen that is harmful to the body , and "self" is synonymous with harmless substances, including all normally functioning cells of the body.

Once pathogens have been detected, the IS must eliminate them in some manner [106]. Different pathogens have to be eliminated in different ways, and we call the cells of the IS that accomplish this *effectors*. The elimination problem facing the immune system is that of choosing the right effectors for the particular kind of pathogen to be eliminated.

5.1.2 The Architecture of the AIS

All discrimination between self and nonself in the IS is based upon chemical bonds that form between protein chains. To preserve generality, we model protein chains as binary strings of fixed length ℓ. The IS must

distinguish self from nonself based on proteins; AIS addresses a similar problem, which we define as follows. The set of all strings of length ℓ. forms a universe, $\mathcal{U}$. which is partioned into two disjoint subsets, which we call self, S. and nonself, $\mathcal{N}$ (i.e. $\mathcal{U} = S \cup \mathcal{N}$, $S \cap \mathcal{N} = \emptyset$) AIS faces a discrimination or classification task: Given an arbitrary string from $\mathcal{U}$ classify it as either normal (corresponding to self) or anomalous (corresponding to nonself).

AIS can make two kinds of discrimination errors: A false positive occurs when a self string is classified as anomalous, and a false negative occurs when a nonself string is classified as normal [107]. The IS also makes similar errors: A false negative occurs when the IS fails to detect and fight off pathogens, and a false positive error occurs when the IS attacks the body (known as an autoimmune response). In the body, both kinds of errors are harmful, so the IS has apparently evolved to minimize those errors; similarly, the goal of AIS is to minimize both kinds of errors. In Figure 5.1 Each string can belong to one of two sets: self or nonself. In this diagram, each point in the plane represents a string; if the point lies within the shaded area it is self, otherwise it is nonself. The immunological detection systemattempts to encode the boundary between the two sets by classifying strings as either normal (corresponding to self) [108] or anomalous (corresponding to nonself).

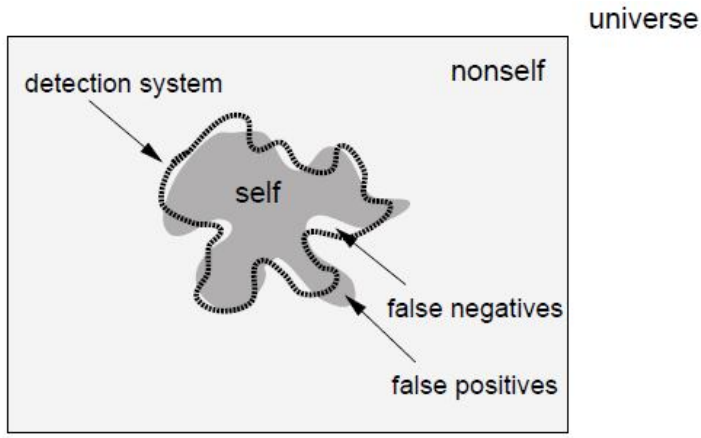


Figure 5.1 A two-dimensional representation of a universe of strings

When real-world problems are mapped to this abstraction, self and nonself may not be disjoint, because some strings may characterize both self and nonself. In this case, the categorization of strings as either one or the other category will lead to unavoidable errors. We do not consider that case here [100]. However, it illustrates the importance of choosing the right characteristic for the application domain: It is essential to choose the equivalent of proteins that can be used to reliably discriminate between self and nonself.

5.1.3 AIS Detectors

Natural immune systems consist of many different kinds of cells and molecules which have been identified and studied experimentally [108]. In our system, we will simplify by introducing one basic type of detector which is modeled on the class of immune cells called *lymphocytes*5. This detector combines properties of B-cells, T-cells, and antibodies. AIS is similar to the IS in that it consists of a multitude of mobile detectors, circulating around a

distributed environment. We model the distributed environment with a graph $G = (V, E)$; each vertex $v \in V$ contains a local set of detectors (called a *detection node*) and detectors migrate from one vertex to the next via the edges. The graph model also provides a notion of locality: Detectors can only interact with other detectors at the same vertex.

Lymphocytes have hundreds of thousands of identical receptors on their surface (and hence are termed *monoclonal*). These receptors bind to regions (epitopes) on pathogens. Binding depends on chemical structure and charge, so receptors are likely to bind to a few similar kinds of epitopes.
The greater the likelihood of a bond occurring, the higher the *affinity* between the receptor and epitope [109]. In AIS, both epitopes and receptors are modeled as binary strings of fixed length ℓ and chemical binding between them is modeled as approximate string matching. In effect, each detector is associated with a binary string, which represents its receptors.
Obvious approximate matching rules include Hamming distance and edit distance, but we have adopted a more immunologically plausible rule, called $r-$ *contiguous bits*: Two strings match if they have r contiguous bits in common as in figure 5.2. The value r is a threshold and determines the specificity of the detector, which is an indication of the size of the subset of strings that a single detector can match [110]. For example, if $r = \ell$ the matching is completely specific, that is, the detector will match only a single string (itself), but if $r = 0$ the matching is completely general, that is, the detector will match every single string of length ℓ

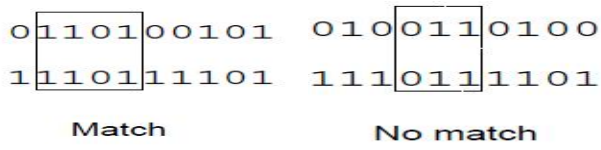


Figure 5.2 Matching under the contiguous bits match rule. In this example, the detector matches for r = 4 but not for r = 3

A consequence of a partial matching rule with a threshold, such as r-contiguous bits, is that there is a trade-off between the number of detectors used, and their specificity—as the specificity of the detectors increases, so the number of detectors required to achieve a certain level of detection also increases. The optimal r is one which minimizes the number of detectors needed, but still gives good discrimination.

A lymphocyte becomes *activated* when its receptors bind to epitopes [111]. Activation changes the state of the lymphocyte and triggers a series of reactions that can lead to elimination of the pathogens.

A lymphocyte will only be activated when the number of its receptors binding to epitopes exceeds a threshold[6]. Chemical bonds between receptors and epitopes are not long-lasting, so to be activated, a lymphocyte must bind sufficient receptors within a short period of time. We model this with *activation thresholds*: A detector must match at least t strings within a given time period to be activated. This is implemented by allowing the detector to accumulate matches, but decaying the match count over time, i.e. there is γ_{match} probability that the match count will be reduced by one at any timestep. This models the probability of a bond between a receptor and an epitope decaying. Once a detector has been activated, its match count is reset to zero [112].

5.1.4 AIS Detectors Training

Lymphocytes are called *negative* detectors because they are trained to bind to nonself; i.e. when a lymphocyte is activated, the IS responds as if nonself were detected. This simple form of learning is known as *tolerization*, because the lymphocytes are trained to be *tolerant* of self. Lymphocytes are created with randomly generated receptors, and so could bind to either self or nonself [113]. One class of lymphocytes, T-cells, is tolerized in a single location, the thymus, which is an organ just behind the breastbone. Immature T-cells develop in the thymus, and if they are activated during development, they die through programmed cell death (*apoptosis*). Most self proteins are expressed in the thymus, so T-cells that survive to maturation and leave the thymus will be tolerant of all those self proteins. This process is called *negative selection*, because the T-cells that are not activated are the ones selected to survive.

Lymphocytes are trained to perform *anomaly* detection. The IS uses a *training* set of self (proteins present in the thymus) to produce detectors that can distinguish between self and nonself.

This clearly will not work if nonself is frequently expressed in the thymus, because then the IS will also be tolerant to that nonself. The underlying assumption is that self occurs frequently compared to nonself. This assumption is the basis of most anomaly detection systems, which define normal as the most frequently occurring patterns or behaviors. AIS uses the *negative selection algorithm*, which is based on negative selection in the IS [114].

In Figure 5.3 The negative selection algorithm. Candidate negative detectors (represented by dark circles) are generated randomly, and if they

match any string in the self set (i.e. if any of the points covered by the detector are in the self set), they are eliminated and regenerated. This process is repeated until we have a set of valid negative detectors that do not match any self strings

The primary difference is that we do not accumulate the self set in a single location, but rather use a form of ansynchronous and distributed tolerization[7]. Each detector is created with a randomly-generated bit string (analogous to a receptor), and remains immature for a time period , called the tolerization period. During this time period, the detector is exposed to the environment (self and possibly nonself strings), and if it matches any bit string it is eliminated [115].

If it does not match during the tolerization period, it becomes a mature detector (analogous to a naive B-cell). Mature detectors need to exceed the match threshold in order to become activated, and when activated they are not eliminated[8], but signal that an anomaly has been detected. Clearly, the assumption here is that if a circulating immature detector matches some self string, it will, with high probability, encounter that self string during its tolerization period, whereas immature detectors thatmatch nonself strings will with low probability encounter those nonself strings during their tolerization period [116].

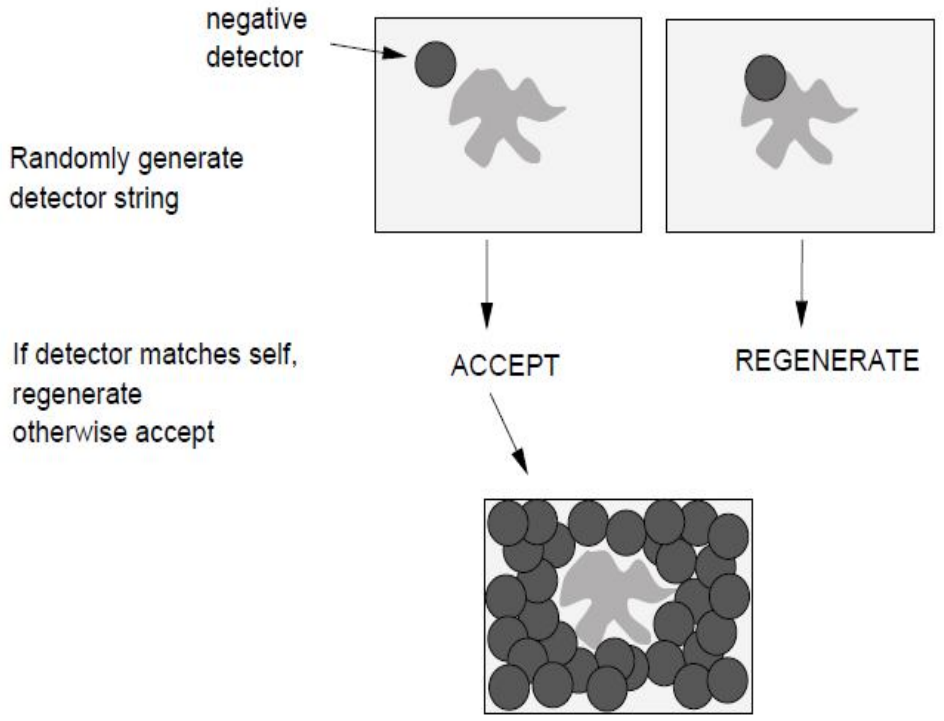


Figure 5.3: The negative selection algorithm

5.1.5 AIS Memory

The IS has an adaptive response that enables it to learn protein structures that characterize pathogens it encounters, and "remember" those structures so that future responses to the same pathogens will be very rapid and efficient. We call this memory-based detection, because the IS "remembers" the structures of known pathogens to facilitate future responses [117]. A memory-based detection system is trained on a subset of nonself to detect particular elements of that subset. When the IS encounters pathogens of a type it has not encountered before, it mounts a *primary response*, which may take several weeks to eliminate the infection; during the primary response the IS is learning to recognize previously unseen foreign patterns.

When the IS subsequently encounters the same type of pathogens, it mounts a *secondary response* which is usually so efficient that there are no clinical indications of a reinfection. The secondary response illustrates the efficacy of memory-based detection.

Memory-based detection in the IS has another important property: It is *associative* [118]. Memory detection allows the IS to detect new pathogens that are structurally related to ones previously encountered. This concept underlies immunization, where innoculation with a harmless form of pathogen, A (such as an attenuated virus) induces a primary response that generates a population of memory cells which are cross-reactive with a harmful kind of pathogen, B. This population of memory cells will ensure that the IS mounts a secondary response to any infections of B.

Primary responses are slow because there may be very few lymphocytes that bind to a new type of pathogen, so the immune response will not be very efficient. To increase efficiency, activated lymphocytes clone themselves, so that there is an exponentially growing population of lymphocytes which can detect the pathogens. The higher the affinity between a lymphocyte's receptors and the pathogen epitopes, the more likely it is that the lymphocyte will be activated [119]. Hence, the lymphocytes that are replicating are those with the highest affinity for the pathogens present. During this time the pathogens are also replicating, so there is a race between pathogen replication and lymphocyte replication. The IS improves its chances in this race through a class of lymphocytes called B-cells, which are subject to high mutation rates (known as somatic hypermutation) when cloning (we currently do not model this aspect). Hypermutation combined with clonal expansion is an adaptive process known as *affinity maturation*. Once the

infection is eliminated, the IS retains a population of *memory* cells: long-lived lymphocytes which have a high affinity for the pathogen. This population of memory cells is of sufficient size and specificity to enable the very rapid secondary response when a reinfection occurs.

AIS uses a similar form of memory-based detection. When multiple detectors at a node are activated by the same nonself string s they enter a competition to become memory detectors. Those detectors that have the closest match (under $r-$ *contiguous bits)* with s will be selected to become memory detectors[9] [120]. These memory detectors make copies of themselves, which then spread out to neighboring nodes. Consequently, a representation of the string s is distributed throughout the graph; future occurrences of s will be detected very rapidly in any node because detectors that match s xist at every node. In addition, memory detectors have lowered activation thresholds (for example, $t = 1$) so that they will be activated far more rapidly in future to reoccurrences of previously encountered nonself strings, i.e. they are much more sensitive to those strings. This mimics the rapid second response seen in the IS.

5.1.6 AIS Sensitivity

A detection event in the IS often results in the production of chemicals (cytokines) which signal other nearby IS cells [121]. To model this, we use the notion of locality inherent in the graph defining the environment for AIS. Each detection node D_i (where i= 1,2….. |V|) has a

local sensitivity level, w_i which models the concentration of cytokines present in a physically local region in the body. The activation threshold of detectors at D_i is defined as $t - w_i$ .e. the higher the local sensitivity, the lower the local activation threshold. Whenever thematch count for amature detector at node $\dot{\iota}$ goes from 0 to 1 the sensitivity level at D_i is increased by 1 [122]. The sensitivity level also has a temporal horizon: over time it decays at a rate given by a decay parameter γ_w which indicates the probability of w_i being reduced by 1. This mechanism ensures that disparate nonself strings will still be detected, providing they occur in a short period of time.

5.1.7 AIS Node Co-stimulation

Unfortunately, tolerization in the IS is not as straightforward, self proteins are never expressed in the thymus, and so lymphocytes that are tolerized centrally in the thymus may bind to these proteins and precipitate an autoimmune reaction. This does not happen in practice because T-cells require *costimulation* to be activated: In addition to binding to proteins (called *signal one*), a T-cell must be costimulated by a *second signal*. This second signal is usually a chemical signal which occurs when the body is damaged in some way. The second signal can come either from cells of the IS or other cells of the body [123]. When a T-cell receives signal one in the absence of signal two, it dies. Hence, autoreactive T-cells (those that bind to self) will be eliminated in healthy tissues. However, if the tissues are damaged, autoreactive T-cells could survive. But they would only survive while the damage persisted; as soon as they left the area of damage, they would receive signal 1 in the absence of signal 2 and die. Moreover, they

would have a high likelihood of dying before they ever reached the area of tissue damage, because of the healthy tissue passed through on the way [124]. Likewise, we cannot assume that in AIS a detector will encounter every string $s \in S$ during its tolerization period, so it is possible that detectors will mature that match some strings in S. Ideally, the second signal should be provided by other components of the system, but our first approximation is to use a human operator to provide the second signal. When a detector d is activated by a string s, it sends a signal to a human operator, who is given a time period T_s (called the costimulation delay) in which to decide if s is really nonself. If the operator decides that s is indeed nonself, a second signal is returned to d If the operator decides that s is actually self, no signal is sent to d and d dies off and is replaced by a new, immature detector. Consequently, a human operator need make no response in the case of false positives; the system will automatically correct itself to prevent similar false positives in future [125].

5.1.8 Detector's cycle of life

If detectors lived indefinitely and only died off when they failed to receive co-stimulation, most detectors would only be immature once. Any nonself strings that occurred during the period of immaturity of these detectors would not be detected in future because all detectors would be tolerant of them and would remain tolerant. In the IS this is not a problem because lymphocytes are typically short-lived (a few days) and so new, immature lymphocytes are always present, i.e. the population of

lymphocytes is dynamic. We introduce a similar measure: Each detector has a probability P_{death} dying once it has matured. When it dies, it is replaced by a new randomlygenerated, immature detector. Ultimately, every detector dies sooner or later, unless it is a memory detector [125].

Figure 5.4 presents the lifecycle of a detector. Now a nonself string will only be undetected if it is continually present to tolerize the continual turnover of new detectors, i.e. the only false negatives will occur when nonself is frequent, which violates a fundamental assumption underlying our model.

An exception to the finite lifespan is memory detectors. In the IS, memory cells are long-lived so that the patterns that they encode are not lost over time. For example, exposure to measles early in life confers life-long protection against the disease. Similarly, memory detectors in AIS are long-lived: they can die only as a consequence of a lack of costimulation. A problem with this mechanism is that eventually all detectors could become memory detectors, with a loss of the advantages conferred by dynamic detector populations [126]. To combat this problem, we limit the number of memory detectors to some fraction m_d of the total detectors. If a new detector wins a competition and becomes a memory detector, and the fraction of memory detectors has reached the limit, then the least-recently-used (LRU) memory detector is demoted to an ordinary mature detector (consequently it once more has a finite lifespan). We demote the LRU detector because the LRU detector is the one that has not been activated for the longest time period of any memory detector, and hence we assume that it is the least useful memory detector.

An additional benefit of a dynamic detector population is that the system can adapt to changing self sets. As the self set changes, it will tolerize new

immature detectors, and mature detectors that were causing false positives will either die from lack of costimulation or from age. Eventually all detectors will be tolerant of self, providing self does not change too quickly [127]. If self changes rapidly compared to the life span of a detector, there will be a sizeable portion of detectors that are immature, because mature detectors will continually die due to lack of costimulation.

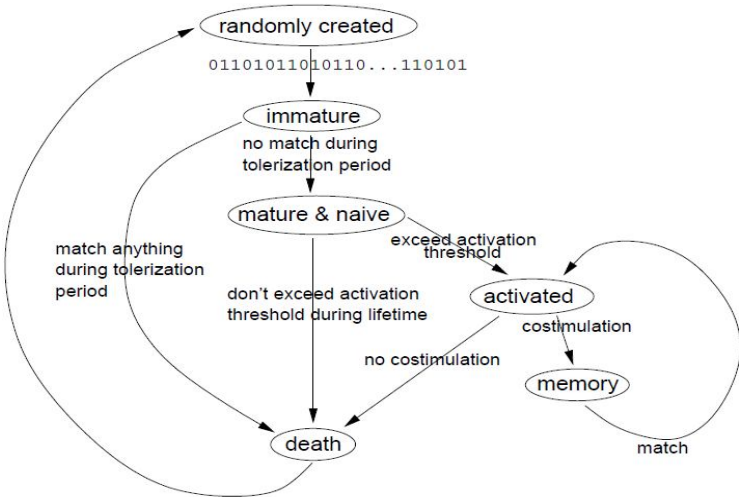


Figure 5.4: lifecycle of a detector

5.1.9 AIS represenation

Molecules of the *major histocompatibility complex* (MHC) play an important role in the IS, because they transport peptides (fragments of protein chains) from the interior regions of a cell and *present* these peptides on the cell's surface. This mechanism enables roving IS cells to detect infections inside cells without penetrating the cell membrane. There are many variations of MHC, each of which binds a slightly different class of peptides. Each individual in a population is genetically capable of making a small set of these MHC types (about ten), but the set of MHC types varies in different individuals. Consequently, individuals in a population are capable

of recognizing different profiles of peptides, providing an important form of population-level *diversity* [128].

MHC plays a crucial role in protecting a population of individuals from *holes* in the detection coverage of nonself. A hole is a nonself string for which no valid detectors can be generated: A nonself string a ϵa N is a hole if and only if, ∀ u ϵa U such that a and u match, then u matches some self string s ϵ S See figure 5.5. Holes can exist for any approximate match rule with a constant probability of matching (such as the r r- contiguous bits), and it is reasonable to assume that they will exist in the biological realm of receptor binding, because binding between receptors in the IS and peptides is approximate. Moreover, pathogens will always be evolving so that they are more difficult to detect (they evolve towards becoming holes in the detection coverage).

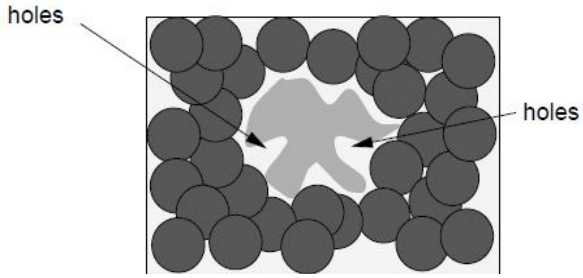


Figure 5.5: self & nonself matching

Those pathogens that are harder to detect will be the ones that survive better and hence are naturally selected. In the IS, each type of MHC can be regarded as a different way of representing a protein (depending on which peptides it presents); in effect, the IS uses multiple representations, or views, of protines [129]. Multiple representations can reduce the overall number of holes, because different representations will induce different holes. In AIS,

each detection node uses a different representation: It filters incoming strings through a randomly-generated permutation mask.

For example, given the strings s_1=01101011, s_2 = 00010011 and a permutation, Λ. Defined by the randomly generated permutation mask 1-6-2-5-8-3-7-4, these strings become $\Lambda(s_1)$= 00111110 and $\Lambda(s_2)$=00001011 Using the contiguous bits rule with $r=3$ s_1 matches s_2 because the last 3 positions are the same, but under the new representation, $\Lambda(s_1)$ does not match $\Lambda(s_2)$ Having a different representation for each detection node is equivalent to changing the "shape" of the detectors, while keeping the "shape" of the self set constant.

This is the situation of as in figure 5.6 in which: Representation changes are equivalent to "shape" changes for detectors. The problem of holes can be ameliorated by using different a representation for each detection node. There are different holes for different representations. or equivalently different shaped detectors can cover different parts of the nonself space, for a global reduction in holes.Consequently, where one node fails to detect a nonself string, another node could succeed.

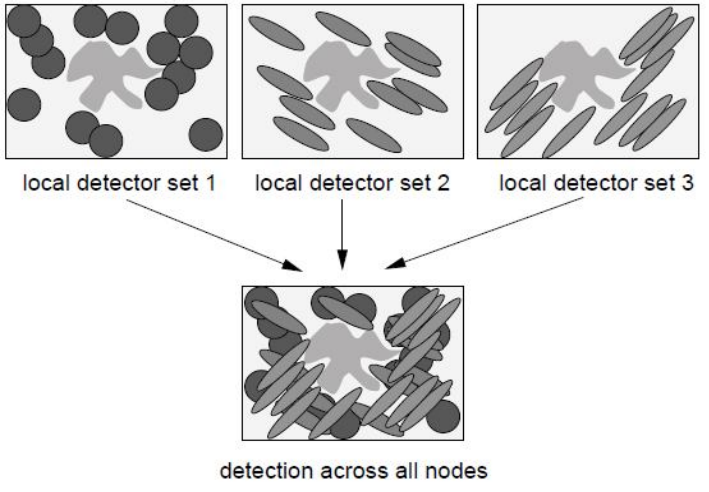


Figure 5.6: Representation of the detection process

5.2 AIS model for Botnets Manipulation in MANET using fuzzy function

In this section, a Security scheme is applied to MANETs in order to discover, handle and treat Botnets. This scheme might be viewed as an immune system that protects and detects any anomaly occurred in the MANET. The detection and protection is based on applying AIS concepts over MANET as will be described later in this section. Instead of using the ordinary AIS method for taking the elimination decision a set of fuzzy rules are used, these rules are rapidly changing in correspondence to the rapidly changing nature of MANET. The rest of this section illustrates the proposed scheme as two stages process.

5.2.1 AIS model:

This section illustrates the usage of the AIS model for providing both innate and gained immunity for the MANET security system

5.2.1.1 The Structure of AIS in MANETs

The security offered by this scheme is based on the efficiency and adaptability provided by the AIS. In this model nodes are viewed as follows:

- The whole set of nodes creates a universe (U) that represents all self and nonself nodes.
- Set of allies (self) nodes (A) .
- Set of enemies (nonself) nodes (E)

These sets must satisfy the following conditions

- $A \cup E = U$ Formulation 1

- $A \cap E = \varnothing$ Formulation 2

To keep generalizing the model the protein chains are modeled as binary short codes. The main purpose of the AIS is to classify a gives set of unknown code into either A or E, to give the decision either to allow or prevent the under classification node.

5.2.1.2 System Sensors:

For simplifying purposes we shall address only the *lymphocyte* cells only. This detector combines properties of B-cells, T-cells, and antibodies. AIS is similar to the IS in that it consists of a multitude of mobile sensors called detectors, circulating around a distributed environment.

Lymphocytes have hundreds of thousands of receptors on its surface (and, therefore, called monoclonal). The pathogen receptor regions (epitopes) are linked.

Charge to the chemical composition and the same type of receptor may bind to epitopes. Binding event, the correlation between the receiver and the high likelihood of epitopes, both epitopes and receptors modeled as a binary string of fixed length L of the chain, the chemical bonds are modeled as approximate matching. In fact, each detector is a binary string, and that is, to its receptors.

A lymphocyte is activated when its receptors accumulated epitopes. Changes in the state of lymphocyte activation and triggers a series of reactions can lead to the elimination of pathogens. Lymphocyte epitopes when left alone for more than a threshold number of receptors bound to be active and survive the elimination process. The chemistry between

receptors and epitopes are not permanent, so active, lymphocyte receptors to bind within a short period of time.

5.2.1.3 Training the System:

The system training is based on the set of fuzzy rules supplied to the system and system negative selection algorithm, which is expressed in the thymus of the symbols to explore collaborative training, based on the maturation of T cells survive and leave the thymus will be tolerant of all proteins that are associated. This process is called negative selection, because the T cells those has not been activated are selected to survive. Each detector is a randomly generated bit string (similar to receptors) are created for a period of time called tolerization period and remains immature. During this time period, the environment (probably associate the enemy strings) and detector are exposed, and if it matches with any bit string then will be eliminated. If it does not match during tolerization, a mature detector (similar to a naive B cell) becomes then it is assumed to be ally.

6.2.1.4 Storage

When multiple receptors at a node are activated by the same Enemy string, they competes to become memory detectors. Those detectors that have the closest match (based on the fuzzy rules) will be selected to become memory detectors. These memory detectors regenerates more copies of themselves, which then spread out to neighboring nodes. Consequently, a representation of the string is distributed throughout the graph; future occurrences of will be detected. In addition, memory detectors have lowered activation thresholds instead of recalling the fuzzy rules again to save time so that they will be activated far more

rapidly in future to reoccurrences of previously encountered Enemy strings, i.e., they are much more sensitive to those strings.

5.2.2 Fuzzy Decision Model (FDM)

The proposed scheme uses Fuzzy Decision function for determining when to activate a T cell this is done by activating the FDM when a new code is noticed.

The fuzzy function is a set of rules based on the following parameters as inputs

1- Memory status (MS): a fuzzy variable that ranges from very week to strong corresponding to the passed string status

2- Number of Nodes (NN): a fuzzy variable that ranges from few to many representing the current number of nodes in the MANET.

The output fuzzy variable "the status of the T-cell" has three fuzzy sets (Pass – Steady – Activate). It should be noted that modifying the membership functions will change the sensitivity of the fuzzy logic system's output to its inputs. Also increasing the number of fuzzy sets of the variables will provide better sensitivity control but also increases computational complexity of the system. Table 6.2.1 show the rules used in the fuzzy logic system.

.

.

Table 5.2: Fuzzy Decision Model

Input		Output
MS	NN	S
Very Week	Few	Pass
Week	Normal	Pass
Strong	Many	Activate
Very Week	Normal	Steady
Week	Many	Pass
Strong	Few	Steady
Very Week	Many	Pass
Week	Many	Activate
Strong	Normal	Activate

Chapter 6: Securing MANETs Using Intutionistic Fuzzy Function as an alternative to negative selection in AIS

In this chapter a new methodology for key distribution in MANETs is presented; that method is based on using the PKI as infrastructure to distribute both session and permanent types of keys. This method may be viewed as a two stages: first an intuitionistic fuzzy model to decide the key length for the current session. Then the key distribution between nodes in MANET both stages are illustrated in the rest of this section.

6.1 Intuitionistic Fuzzy Sets

A fuzzy set is a nebular collection of elements from a universe M described by and identified with a (membership) function A: $M \rightarrow [0, 1]$ [10]. An *intuitionistic fuzzy set* is instead a nebular collection of elements from M identified with a pair $\mathscr{E}(A, A^d)$, where

$$A, A^d: M \rightarrow [0, 1] \text{ and } \forall x \in M: A(x) + A^d (x) \leq 1.$$

one interprets A as a membership function: $A(x)$ is a degree of membership of x in the intuitionistic fuzzy set $\mathscr{E}$, whereas A^d, a function *dual* to A, is understood as a *non-membership function*, i.e. $Ad(x)$ does express a *degree of non-membership* of x in that intuitionistic fuzzy set. Finally the term $\chi_{\mathscr{E}}$ in the following equation is called the degree of *hesitation* whether or not x is in $\mathscr{E}$

$$\chi_{\mathscr{E}}(x) = 1- (A+ A^d) \text{ [15].}$$

6.2 Intuitionistic fuzzy model for Key Size Determination Function

The security offered by the algorithm is based on the difficulty of discovering the secret key through a brute force attack. Mobile Status (MS) Security Level is the correlative factor being analyzed with three considerations:

1- The longer the password, harder to withstand a severe attack of brute force. In this research the key lengths from 16 to 512 are assumed

2- The quickest way to change passwords, more secure the mobile host. It is more difficult to decipher the key to a shorter time. A mobile host to change the secret key is often safer than a mobile host using a constant secret key.

3- The neighbor hosts the mobile host has, the more potential attacker. I.e. the possibility of attack is greater. There are many other factors affecting the safety of mobile hosts, such as bandwidth. The security level of mobile hosts is a function with multiple variables and affected more than one condition.Here a intuitionistic fuzzy logic system is defined. Inputs of the intuitionistic fuzzy logic system are the frequency of changing keys *(f)* and the number of neighbor hosts *(n)*. Output of the intuitionistic fuzzy logic system is the Security-Level of M*S*. It is assumed that the three factors are independent with each other. The relationship of them is as follows:

$$S \propto l \bullet f \bullet \frac{1}{n}. \qquad \text{Formula 1}$$

It means that the Security-Level of MH is in direct proportion to the length of the key and the frequency of changing keys, in inverse proportion to the number of neighbor hosts. The *S* value is updated by

the intuitionistic fuzzy logic system. When the key length is short, the Security-Level of MH should be low; otherwise the Security-Level of MS should be high.

- The first input parameter to the intuitionistic fuzzy variable "the number of neighbor hosts" has three intuitionistic fuzzy sets—few, normal and many. The membership function of *n* is illustrated in Figure 6.1.

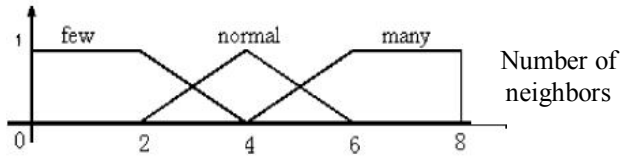


Figure 6.1: Membership function of intuitionistic fuzzy variable *n*.

- The input intuitionistic fuzzy variable "the frequency of changing keys" has two intuitionistic fuzzy sets—slow and fast. The membership functions of *f* is showed in formulation (2)

$$f = \begin{cases} slow & \text{the secret key is constant} \\ fast & \text{the secret key is variable.} \end{cases}$$ Formula 2

- The output intuitionistic fuzzy variable "the Security-Level of MS" has five intuitionistic fuzzy sets containing the set and its complementary set. These sets are(lowest, low, normal, high and highest). It should be noted that modifying the membership functions will change the sensitivity of the intuitionistic fuzzy logic system's output to its inputs. Also increasing the number of intuitionistic fuzzy sets of the variables will provide better sensitivity control but also increases computational complexity of the system. Table 6.1 show the rules used in the intuitionistic fuzzy logic system.

Table 6.1: the intuitionistic fuzzy system rules

Input		Output
F	N	S
Slow	Few	$\mathscr{E}$ (Low , ~Low)
Slow	Normal	$\mathscr{E}$ (Lowest , ~Lowest)
Slow	Many	$\mathscr{E}$ (Lowest , ~ Lowest)
Fast	Few	$\mathscr{E}$ (Normal , ~ Normal)
Fast	Normal	$\mathscr{E}$ (Low , ~ Low)
Fast	Many	$\mathscr{E}$ (Low , ~Low)
Slow	Few	$\mathscr{E}$ (High , ~High)
Slow	Normal	$\mathscr{E}$ (Normal , ~ Normal)
Slow	Many	$\mathscr{E}$ (Low , ~ Low)
Fast	Few	$\mathscr{E}$ (Highest, ~ Highest)
Fast	Normal	$\mathscr{E}$ (High , ~ High)
Fast	Many	$\mathscr{E}$ (High , ~ High)

The output of that system determines the number of bits used and the security level required for the current situation varying the number of bits between 16 and 256 bits. This determination is based on the IFS analysis whish passes the two parameters (A, A^d) then based on that analysis the system decides the accurate key size in each situation

6.3 key distribution

Once the intuitionistic fuzzy function has decided the length of the session key based on its criteria the problem of key creation and distribution arises. The nature of NANET poses great challenges due to the lake of infrastructure and control over the network. To overcome such problems the use of PK scheme is used to distribute the key under the assumption that one node (let us say the first node that originates the network) is responsible for the creation of session keys. If that node is going to leave the network it must transfer the process of key creation to another trusted node in the network.

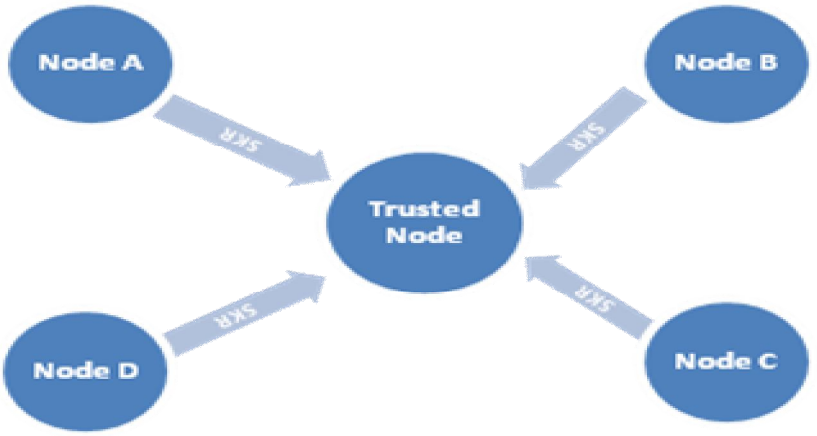
1- Each node sends a message (Session Key Request SKR) encrypted with its private key (that message contains a key request and a timer) to the key creator node which owns a table that contains the public key for each node in the network. Figure 6.2 (a) where the direction of the arrow's head denotes the private key used encryption is the originating node.

2- The key creator node simply decrypts the message and retrieves the request and the timer with one of the following scenarios occurs:

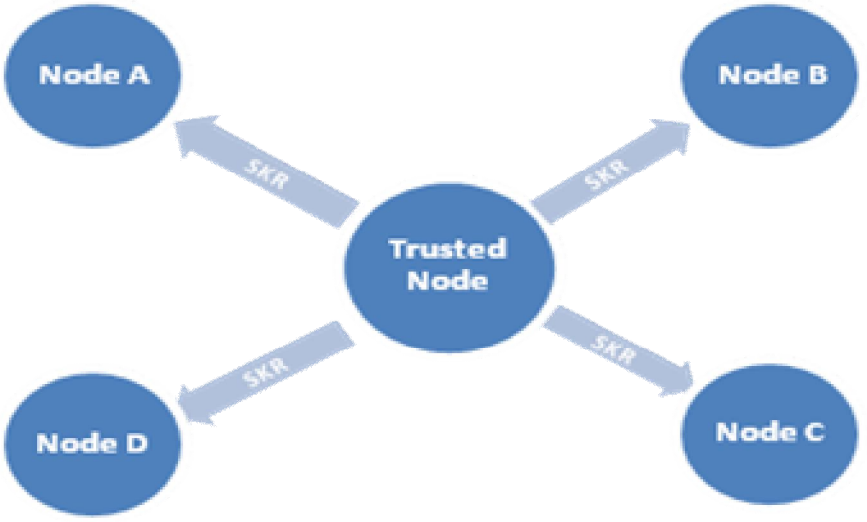
3- The timer was expired or the message is unreadable the message is neglected.

4- The timer is valid and the decryption of the message using the corresponding Public Key gives a readable request. The key creator node sends a message to that node containing the current session key. That message is encrypted two times first using the key creator's

Private key (for authentication) then using the destination's public key Figure 6.2 (b). Where the direction of the arrow's head denotes the private key used encryption is the trusted node then with the destination node's Public Key.



(a)



(b)

Figure 6.2 key distribution : (a) SK Request (b)SK Response

5- Any time the intuitionistic fuzzy model reports that the network condition changes; the key creator node sends a jamming message for every node currently in the network asking them to send a key request message.

6- Any authenticated node (including the Trusted node) on the network knowing the current session key can send messages either to every node or to a single node on the network, simply by encrypting the message using the current session key.

6.4 AIS System Components:

This section illustrates the usage of the AIS as a system that provides both types of immunity (Innate & Gained) for the MANET security system. This system acts as a permanent defense wall in the face of any misbehavioring node. In this technique the intuitionistic fuzzy function has replaced the ordinary negative selection algorithm as the results will show.

6.4.1 The Structure of AIS in MANETs

Self and nonself may not be disjoint, because some strings may characterize both self and nonself. In this case, the categorization of strings as either one or the other category will lead to unavoidable errors. We do not consider that case here. However, it illustrates the importance of choosing the right characteristic for the application domain: It is essential to choose the equivalent of proteins that can be used to reliably discriminate between self and nonself.

The security offered by this scheme is based on the efficiency and adaptability provided by the AIS. In this model nodes are viewed as follows:

- The whole set of nodes creates a universe (U) that represents all self and nonself nodes.
- Set of allies (self) nodes (A) .
- Set of enemies (nonself) nodes (E)

These sets must satisfy the following conditions

- $A \cup E = U$ Formulation 1
- $A \cap E = \varnothing$ Formulation 2

To keep generalizing the model the protein chains are modeled as binary short codes. The main purpose of the AIS is to classify a gives set of unknown code into either A or E, to give the decision either to allow or prevent the under classification node.

6.4.2 System Sensors:

For simplifying purposes we shall address only the *lymphocyte* cells only. This detector combines properties of B-cells, T-cells, and antibodies. AIS is similar to the IS in that it consists of a multitude of mobile sensors called detectors, circulating around a distributed environment.

Lymphocytes have hundreds of thousands of receptors on its surface (and, therefore, called monoclonal). The pathogen receptor regions (epitopes) are linked.

A sensor must match at least τ strings within a given time period to be activated. This is implemented by allowing the detector to accumulate matches, but decaying the match count over time, i.e. there is γ_{match}

probability that the match count will be reduced by one at any timestep. This models the probability of a bond between a receptor and an epitope decaying. Once a detector has been activated, its match count is reset to zero. Charge to the chemical composition and the same type of receptor may bind to epitopes. Binding event, the correlation between the receiver and the high likelihood of epitopes, both epitopes and receptors modeled as a binary string of fixed length L of the chain, the chemical bonds are modeled as approximate matching. In fact, each detector is a binary string, and that is, to its receptors.

The sensor is activated when its receptors accumulated epitopes. Changes in the state of lymphocyte activation and triggers a series of reactions can lead to the elimination of pathogens. Lymphocyte epitopes when left alone for more than a threshold number of receptors bound to be active and survive the elimination process. The chemistry between receptors and epitopes are not permanent, so active, lymphocyte receptors to bind within a short period of time.

6.4.3 Sensor's Life Cycle:

Figure 6.4 illustrates the life cycle of the system's sensors

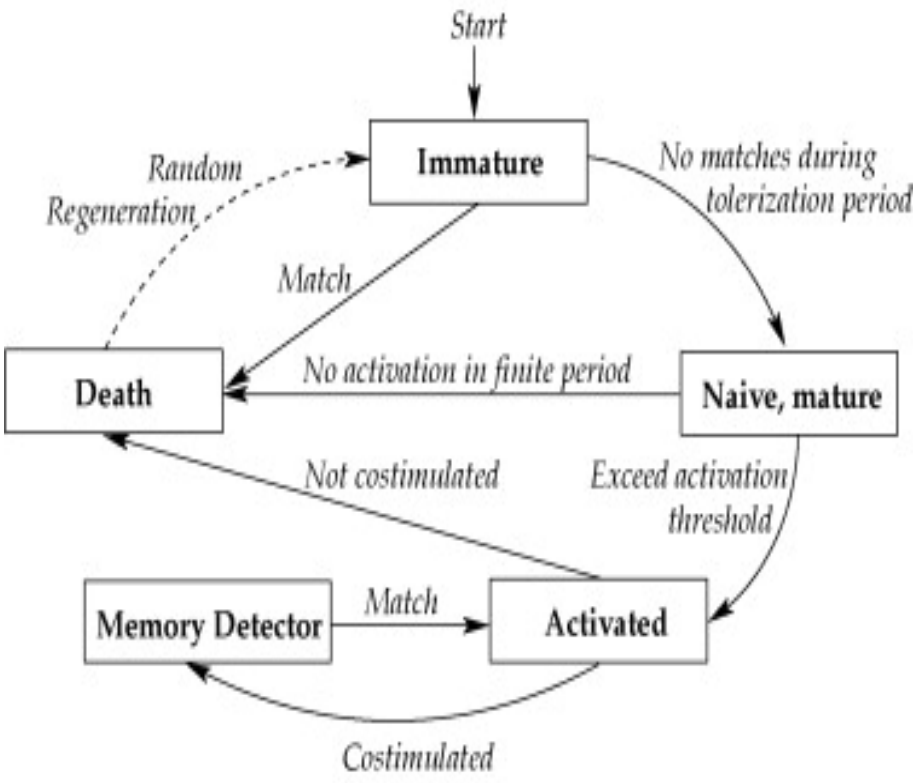


Figure 6.4: Sensor's Life Cycle

In figure 6.4 presents the lifecycle of a detector. Now a nonself string will only be undetected if it is continually present to tolerize the continual turnover of new detectors, i.e. the only false negatives will occur when nonself is frequent, which violates a fundamental assumption underlying our model.

An exception to the finite lifespan is memory detectors. In the IS, memory cells are long-lived so that the patterns that they encode are not lost over time. For example, exposure to measles early in life confers life-long protection against the disease. Similarly, memory detectors in AIS are long-lived: they can die only as a consequence of a lack of costimulation. A problem with this mechanism is that eventually all detectors could become

memory detectors, with a loss of the advantages conferred by dynamic detector populations. To combat this problem, we limit the number of memory detectors to some fraction m_d of the total detectors. If a new detector wins a competition and becomes a memory detector, and the fraction of memory detectors has reached the limit, then the least-recently-used (LRU) memory detector is demoted to an ordinary mature detector (consequently it once more has a finite lifespan). We demote the LRU detector because the LRU detector is the one that has not been activated for the longest time period of any memory detector, and hence we assume that it is the least useful memory detector.

.

An additional benefit of a dynamic detector population is that the system can adapt to changing self sets. As the self set changes, it will tolerize new immature detectors, and mature detectors that were causing false positives will either die from lack of costimulation or from age. Eventually all detectors will be tolerant of self, providing self does not change too quickly. If self changes rapidly compared to the life span of a detector, there will be a sizeable portion of detectors that are immature, because mature detectors will continually die due to lack of costimulation.

6.4.4 Training the System:

The system training is based on the set of fuzzy rules supplied to the system and system negative selection algorithm, which is expressed in the thymus of the symbols to explore collaborative training, based on the maturation of T cells survive and leave the thymus will be tolerant of all proteins that are associated. This process is called negative selection,

because the T cells those has not been activated are selected to survive. Each detector is a randomly generated bit string (similar to receptors) are created for a period of time called tolerization period and remains immature. During this time period, the environment (probably associate the enemy strings) and detector are exposed, and if it matches with any bit string then will be eliminated. If it does not match during tolerization, a mature detector (similar to a naive B cell) becomes then it is assumed to be ally.

6.4.4 Storage

When multiple receptors at a node are activated by the same Enemy string, they competes to become memory detectors. Those detectors that have the closest match (based on the fuzzy rules) will be selected to become memory detectors. These memory detectors regenerates more copies of themselves, which then spread out to neighboring nodes. Consequently, a representation of the string is distributed throughout the graph; future occurrences of will be detected. In addition, memory detectors have lowered activation thresholds instead of recalling the fuzzy rules again to save time so that they will be activated far more rapidly in future to reoccurrences of previously encountered Enemy strings, i.e., they are much more sensitive to those strings.

6.5 Inter-Nodes Communications

After each node is supposed to be authenticated by an intermediate trusted node, it can send messages to any other desired node as illustrated in figure 6.5

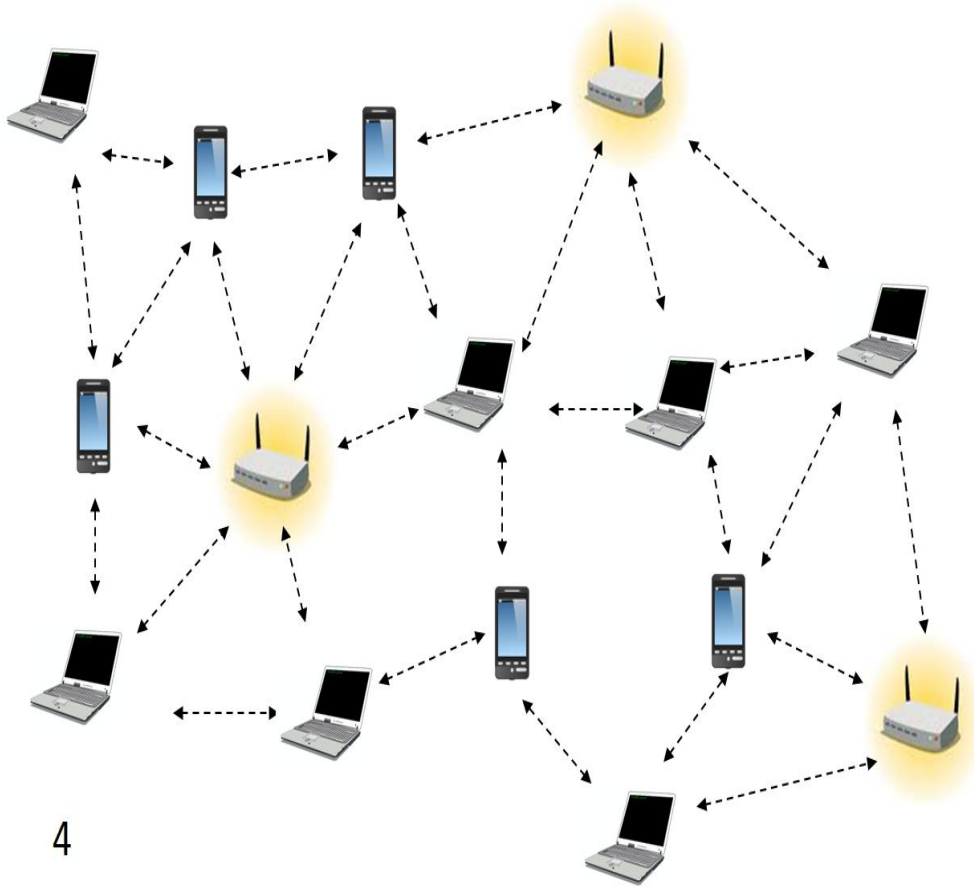


Figure 6.5 inter-node communications after verifiction

6.6 Public Key Security

The distinctive technique used in public key cryptography is the use of asymmetric key algorithms, where the key used to encrypt a message, not the same as the key used to decrypt it. Each user has a pair of cryptographic keys - a public encryption key and a private decryption key . The provision

of public key cryptography is widely distributed, while the private-decryption key is known only to the recipient. Messages are encrypted with the recipient's public key and can only be decrypted with the corresponding private key .

The keys are mathematically related, but the parameters are chosen so that the determination of the private key of the public key is prohibitively expensive. The discovery of algorithms that can produce pairs of public / private key revolutionized the practice of cryptography in principle in mid-1970.

In contrast, symmetric key algorithms, variations of which have been used for thousands of years, uses a single secret key - that should be shared and kept private by the sender and receiver - for encryption and decryption. To use a symmetric encryption scheme, the sender and receiver must share the key securely in advance.

Because symmetric key algorithms are almost always much less computationally intensive, it is common to exchange a key using a key exchange algorithm and transmit data using that key and symmetric key algorithm. Family PGP and SSL / TLS schemes do this, for example, and therefore speak of hybrid cryptosystem.

- The two main branches of public key cryptography are:

Public Key Encryption: a message encrypted with the recipient's public key can be decrypted by anyone except a holder of the corresponding private key - presumably this will be the owner of that key and the person associated with the public key used. This is used for confidentiality.

• Digital signatures (Authentication): a signed message with the sender's private key can be verified by anyone with access to the sender's public key, which shows that the sender had access to the private key (and therefore likely to be the person associated with the public key used), and part of the message has not been tampered with. On the question of authenticity, see also the summary of the message.

The main idea behind public-key (or asymmetric) cryptosystems is the following:

One entity has (in contrast to symmetric cryptosystems) a pair of keys which are called the private key and the public key. These two parts of the key pair are always related in some mathematical sense. As for using them, the owner of such a key pair may publish her public key, but it is crucial that she keeps the private key only for herself. Let (sk, pk) be such a key pair where sk is the Secret private Key for node (A) and pk is the corresponding public key . If a second node wants to securely send a message to (A) it computes:

C = encrypt(M, pk) where encrypt denotes the so-called encryption function which is also publicly known .

This function is a one-way function with a trap-door. In other words, the trap-door allows for the creation of the secret key sk which in turn enables Alice to easily invert the encryption function. We call C the ciphertext. Obtaining M from C can be done easily using the (publicly known) decryption function decrypt and A's private key (sk). On the other hand, it is much harder to decrypt without having any knowledge of the private key. As already mentioned, the great advantage of this approach is that no secure key exchange is necessary before a message is transmitted.

Chapter 7: Experimental results and conclusions

This chapter provides the set of experimental records achieved while attempting to create security models to handle security issues in MANETs

7.1 Security in MANET based on PKI using fuzzy function

In this Section the set of experimental results for the attempts to decide the way for creating a more secured MANETs. These experiments are clarified.

7.1.1 Fuzzy vs. Non-Fuzzy Key size determination function:

The first type of experiments had taken place to decide the key size for the encryption process. To accomplish this job the ordinary mechanism of KNN is used as a non-fuzzy technique. Given the same parameters passed to the fuzzy and the non-fuzzy function the performance is measured with evaluation criteria are the average security-level and the key creation time.

The performance criteria are demonstrated in the following sections:

7.1.1.1 The Average security-level:

Average security level is measured for both techniques as the corresponding key provided how much strength given the number of nodes, the results are scaled from 0 to 5 these results are shown in table 7.1 and figure 7.1

Table 7.1 ASL of fuzzy vs. non-fuzzy classification

No. nodes	25	50	75	100	125	150	175	200	225	250
Non-Fuzzy	2.6	2.1	2.5	2.2	1.5	1.7	1.4	2.3	2	1.5
FC	3.4	3.6	3.8	3.9	4	4	4	4	4	4

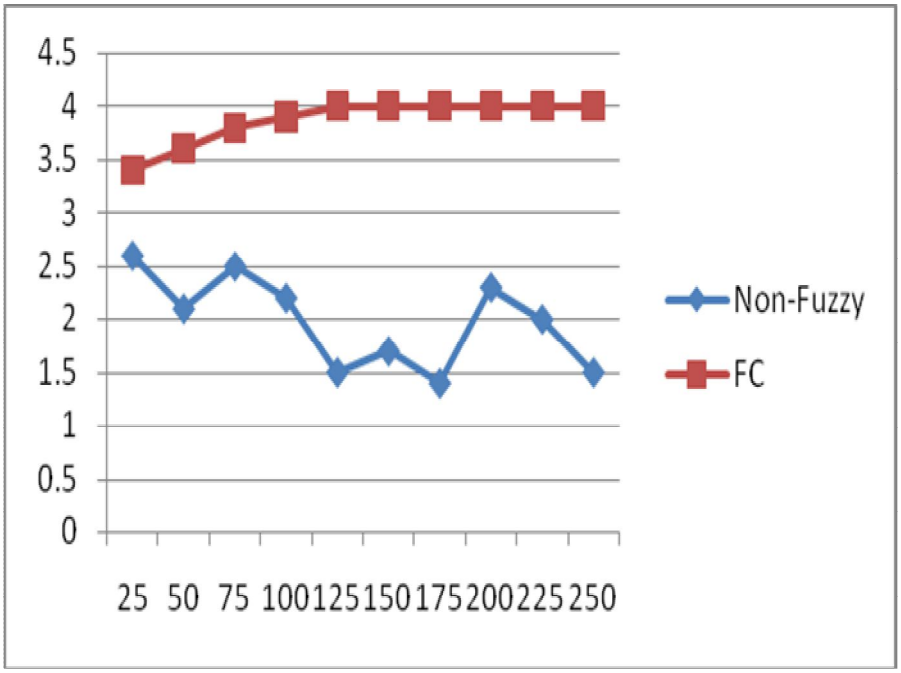


Figure 7.1: average security-level vs the number of mobile nodes

Figure 7.1. and table 7.1 shows the average security level with the number of mobile nodes between 25 and 250. As shown in the figure and the table, the average security-level of the Fuzzy Classifier (FC) is much higher than the average security-level of the non-fuzzy classifier, especially for many

mobile nodes. This is an expected result since the fuzzy classifier adapts its self upon the whole set of criteria.

7.1.1.2 The key creation time:

The time required to generate the key in both cases are measured, the results are scaled from 0 to 1 and are shown in table 7.2 and figure7.2

Table 7.2: KCR of fuzzy vs. non-fuzzy classifiers

No. nodes	25	50	75	100	125	150	175	200	225	250
Non-Fuzzy	0.95	0.93	0.95	0.96	0.96	0.96	0.96	0.96	0.96	0.96
FC	0.93	0.9	0.85	0.92	0.93	0.94	0.94	0.94	0.94	0.94

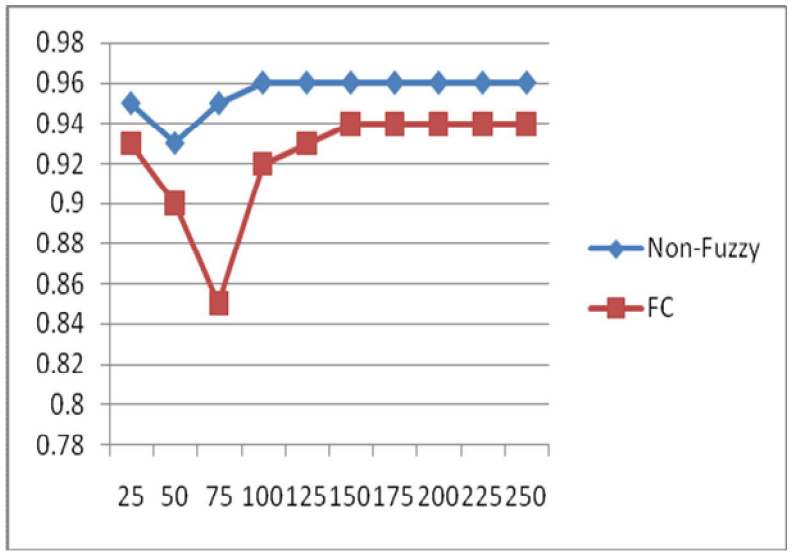


Figure 7.2: Key creation time vs. the number of mobile nodes.

Figure 7.2 and table 7.2 shows the Key creation time with the number of mobile nodes between 25 and 250. The speed of Key creation is very high (mostly above 0.94) for all two techniques. However, the Non-fuzzy technique has some faster Key creation time than the Fuzzy Classifier, especially with few mobile nodes. The reason is that the smaller the number of nodes with the same amount of calculation the bigger the time taken.

7.1.2 PKI vs. non-PKI distribution

After the Key size had been determined via the Key size determination function the final problem is to distribute that key among nodes on the network. There were two approaches for the key distribution problem either PKI or non-PKI. In this subsection the results of applying PKI and non-PKI techniques is illustrated as applied in terms of security and processing time

7.1.2.1 Security of PKI vs. Non PKI

The PKI presents more overall security than ordinary non-PKI (single key) that is illustrated by applying both techniques over the network and recording the results regarding to the time required for an external attacker to break the session key. Table 7.3 and figure 7.3 shows that results under the assumption of using small public-private key pairs

Table 7.3: security of PKI vs, non-PKI

No. nodes	25	50	75	100	125	150	175	200	225	250
Non-PKI	0.15	0.2	0.23	0.26	0.3	0.32	0.36	0.4	0.44	0.45
PKI	0.8	0.85	0.85	0.92	0.93	0.94	0.94	0.94	0.94	0.94

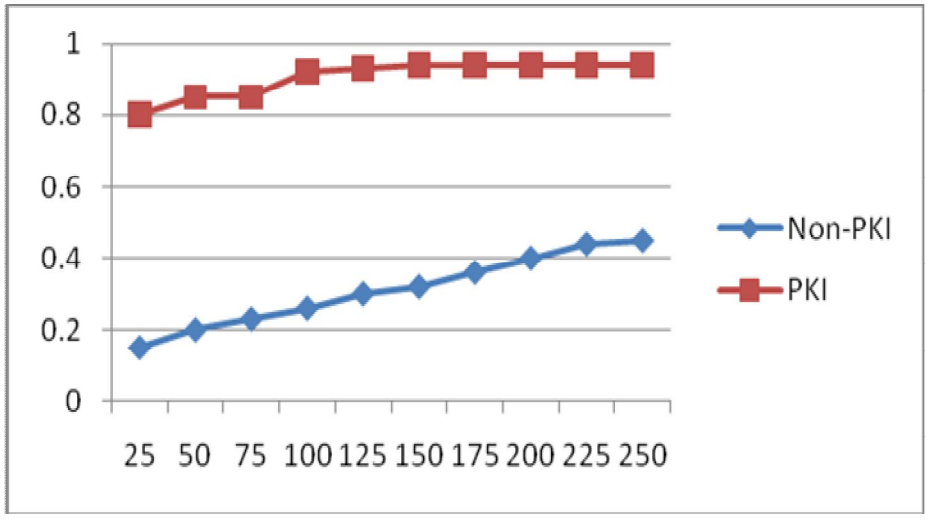


Figure 7.3: security of PKI vs, non-PKI

In graph and figure shows the huge difference in the security level provided by the PKI technique over the Non-PKI mechanism given the same experimental conditions.

7.1.2.2 Processing time of PKI vs. Non PKI

Another factor had been taken into consideration while developing the model that is time required to process the key and distribute it. Table 7.4

and figure 7.4 shows that results under the assumption of using small public-private key pairs

Table 7.4: Processing time of PKI vs. non-PKI

No. nodes	25	50	75	100	125	150	175	200	225	250
Non-PKI	0.3	0.32	0.35	0.37	0.4	0.44	0.47	0.51	0.55	0.58
PKI	0.2	0.35	0.5	0.6	0.68	0.75	0.83	0.87	0.93	0.97

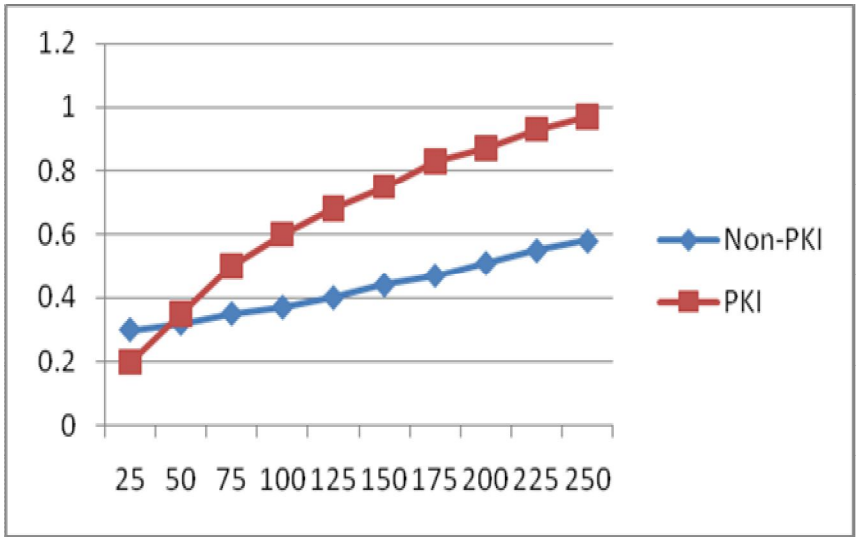


Figure 7.1.4: Processing time of PKI vs. non-PKI

Table 7.4 and the Figure 7.4 shows that Non-PKI techniques provides relatively small amount of processing time than PKI this due to the amount of modular arithmetic performed in the PKI mechanisms. However the difference in the processing time is neglectable comparing to the security level provided by the PKI under the same conditions

7.2 AIS model for Botnet Detection in MANET using fuzzy function

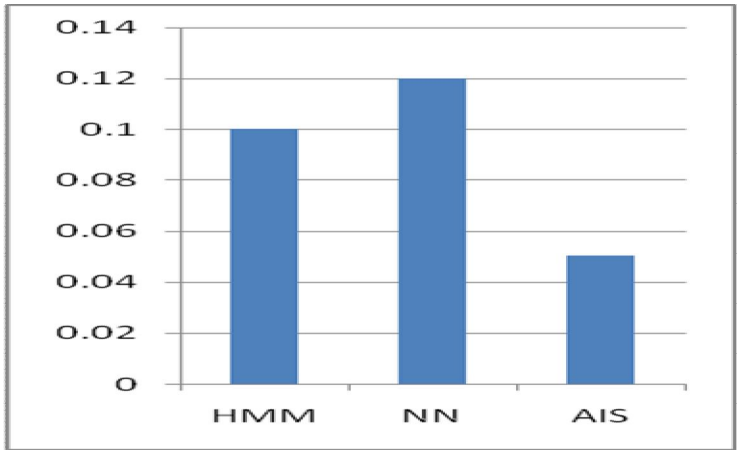
In this section the set of experimental results for the attempts to decide the way for creating a more secured MANETs to handle a dangerous attack like Botnets . These experiments are clarified.

7.2.1 AIS vs. Hidden Markov Models (HMMs) and Neural Networks (NNs):

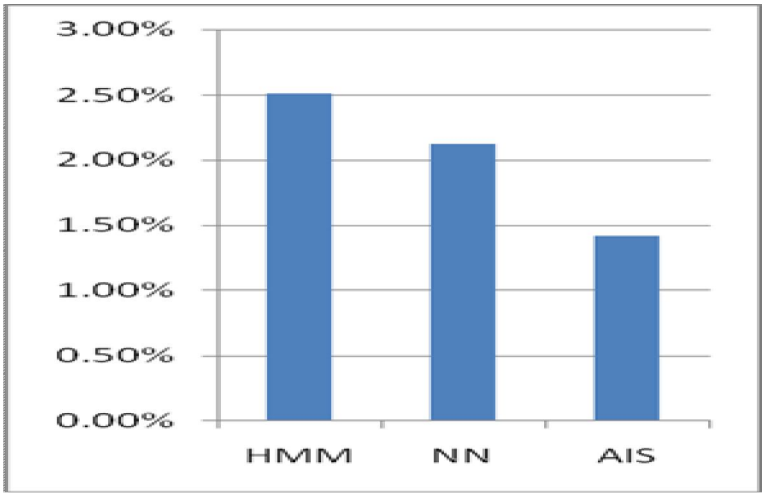
At this point in the research the AIS is compared to two highly used techniques (HMMs) and (NNs).

Table 7.5 results of applying AIS vs NNs and HMMs

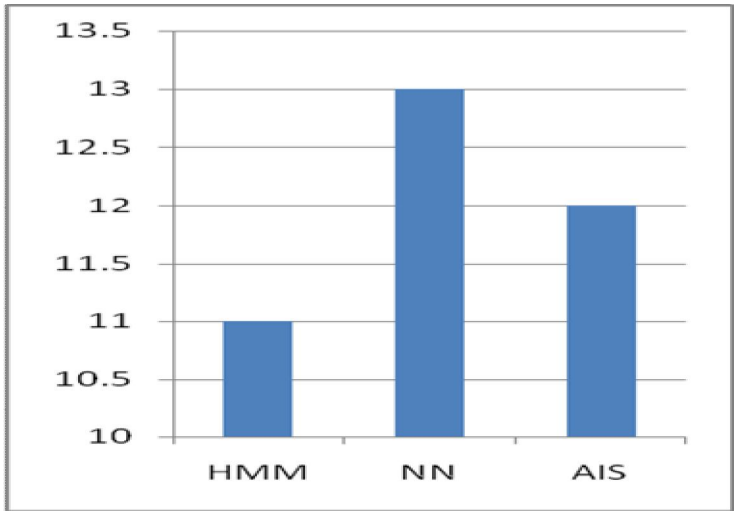
Classification system	Error Rate	Average Decision time	Average Training time
HMM	2.51 %	0.10 sec	11 minutes
NN	2.12 %	0.12 sec	13 minutes
AIS	1.42 %	0.05	12 minutes



(a)



(b)



(c)

Figure 7.5 results of applying AIS vs NNs and HMMs

(a) Error rates (b) Average Decision Time (c) Average Training Time

As table 7.5 and figure 7.5 illustrates the superiority of the AIS over both HMMs and NNs that is AIS provides relatively less error rate and small decision time with an obvious small training time.

7.2.2 Fuzzy vs. Negative selection Decision:

Another important type of experiments had taken place to decide the action of the T-Cells. To assure that the proposed mechanism works better, the ordinary negative selection mechanism is compared to the proposed fuzzy function. The performance is measured with two evaluation criteria

1- The Decision correctness and 2- The Decision time.

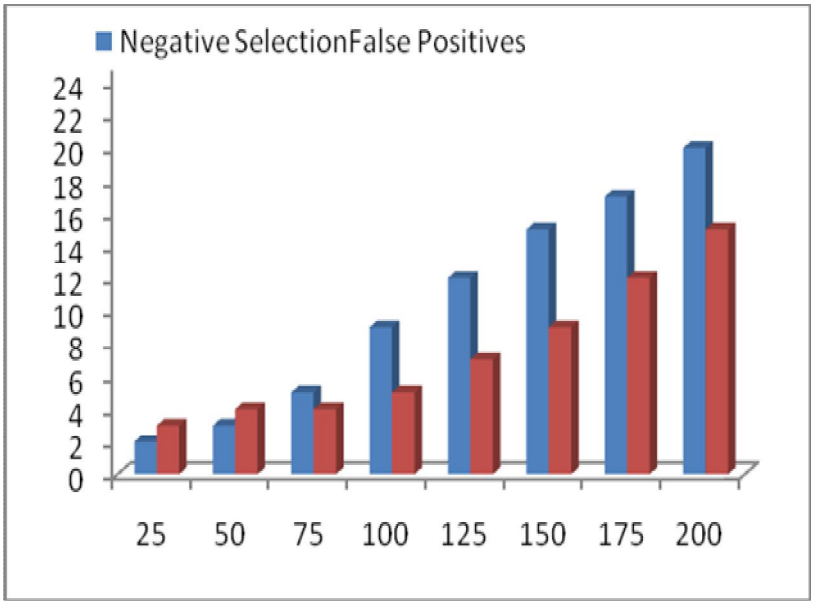
The performance criteria are demonstrated in the following sections:

7.2.2.1 The Decision correctness

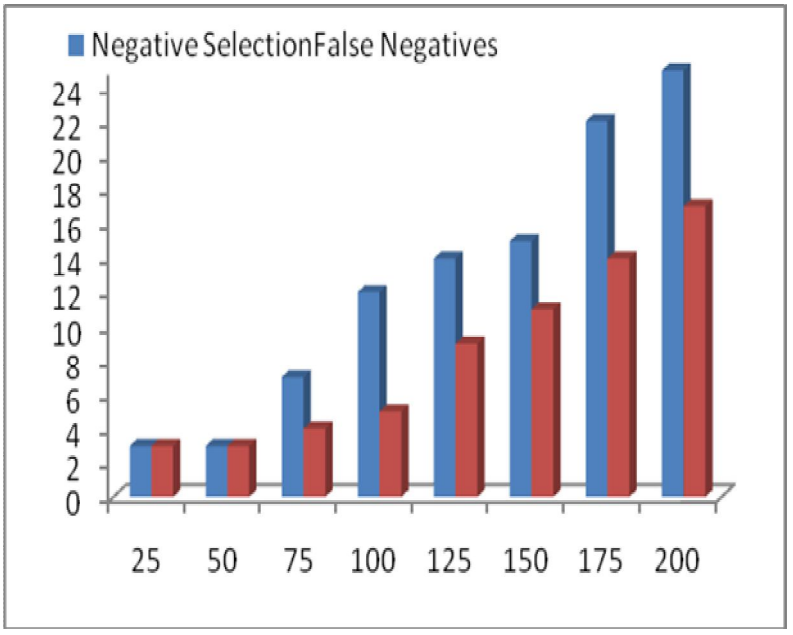
The Decision correctness is measured for both techniques. The measurement process is based on monitoring both techniques and counting the set of false positives and negatives for a set of nodes varying from 25 to 200.

Table 7.6 False Positive Decisions for fuzzy vs. –ve selection

	Negative Selection		Fuzzy Decision	
No. nodes	False	False	False	False
25	2	3	3	3
50	3	3	4	3
75	5	7	4	4
100	9	12	5	5
125	12	14	7	9
150	15	15	9	11
175	17	22	12	14
200	20	25	15	17



(a)



(b)

Figure 7.6: average security-level vs. the number of mobile nodes

(a) False positive decisions (b) False negative decisions

Figure 7.6 and table 7.6 shows that the error occurred by the Fuzzy Decision Function is remarkably small than the Negative selection mechanism. Even though in the small number of nodes both mechanisms show almostly the same results but the overall performance of the fuzzy decision function is higher.

7.2.2.2 The Decision time:

The time required to take the decision either to eliminate the node or not in both cases are measured, the results are shown in table 7.7 and figure 7.7 both results are in seconds.

Table 7.7: Decision time of –ve selection vs. fuzzy function

Number of nodes	Negative Selection	Fuzzy
25	0.003	0.008
50	0.007	0.01
75	0.01	0.015
100	0.15	0.017
125	0.019	0.02
150	0.023	0.021
175	0.027	0.025
200	0.032	0.029
225	0.04	0.033
250	0.048	0.037

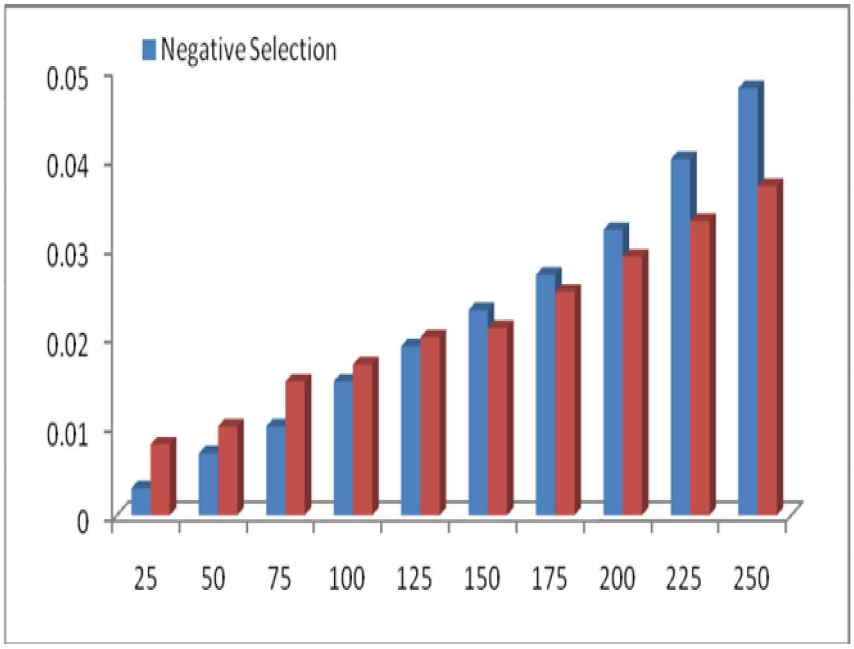


Figure 7.7: Decision time of –ve selection vs. fuzzy function

Figure 7.7 and table 7.7 shows the Key creation time with the number of mobile nodes between 25 and 250. The speed of Decision time is very small for all two techniques. However, the fuzzy decision rules have faster Key decision time than the negative selection, especially with many mobile nodes.

7.3 A PKI based Security Model for MANET using Intuitionistic Fuzzy Function

In this section the set of experimental results for the attempts to decide the way for creating a more secured MANETs. These experiments are clarified.

7.3.1 Intuitionistic fuzzy vs. Non-Intuitionistic fuzzy Key size determination function

The first type of experiments had taken place to decide the key size for the encryption process. To accomplish this job the ordinary mechanism of KNN is used as a non-intuitionistic fuzzy technique. Given the same parameters passed to the intuitionistic fuzzy and the non-intuitionistic fuzzy function the performance is measured with evaluation criteria are the average security-level and the key creation time.

The performance criteria are demonstrated in the following sections:

7.3.1.1 The Average security-level

Average security level is measured for both techniques as the corresponding key provided how much strength given the number of nodes, the results are scaled from 0 to 5 these results are shown in table 7.8 and figure 7.8:

Table 7.8 ASL of intuitionistic fuzzy vs. non-intuitionistic fuzzy classification

Number of nodes	Non-Intuitionistic FC	Intuitionistic FC
25	2.6	3.4
50	2.1	3.6
75	2.5	3.8
100	2.2	3.9
125	1.5	4
150	1.7	4
175	1.4	4
200	2.3	4
225	2	4
250	1.5	4

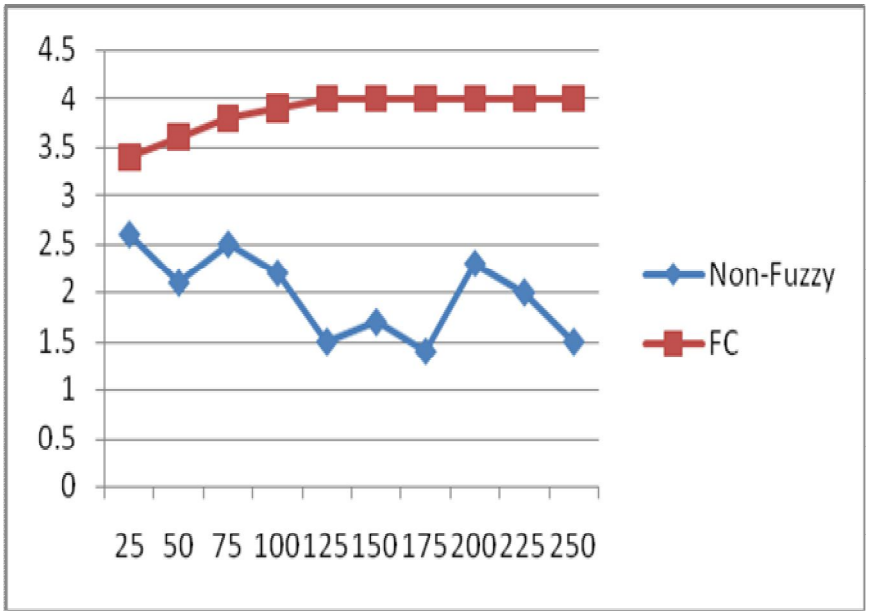


Figure 7.8: average security-level vs the number of mobile nodes

Figure 7.8 and table 7.8 shows the average security level with the number of mobile nodes between 25 and 250. As shown in the figure and the table, the average security-level of the Intuitionistic fuzzy Classifier (FC) is much higher than the average security-level of the non-intuitionistic fuzzy classifier, especially for many mobile nodes. This is an expected result since the intuitionistic fuzzy classifier adapts its self upon the whole set of criteria.

7.3.1.2 The key creation time

The time required to generate the key in both cases are measured, the results are scaled from 0 to 1 and are shown in table 7.9 and figure 7.9

Table 7.9: KCR of intuitionistic fuzzy vs. non-intuitionistic fuzzy classifiers

Number of nodes	Non-Intuitionistic FC	Intuitionistic FC
25	.095	0.93
50	0.93	0.9
75	0.95	0.85
100	0.96	0.92
125	0.96	0.93
150	0.96	0.94
175	0.96	0.94
200	0.96	0.94
225	0.96	0.94
250	0.96	0.94

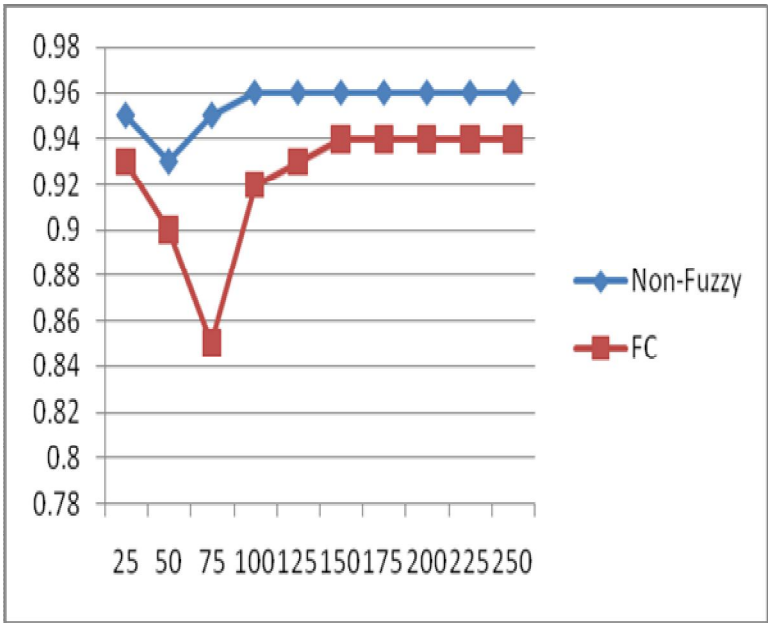


Figure 7.9: Key creation time vs the number of mobile nodes.

Figure 7.9 and table 7.9 shows the Key creation time with the number of mobile nodes between 25 and 250. The speed of Key creation is very high (mostly above 0.94) for all two techniques. However, the Non-intuitionistic fuzzy technique has some faster Key creation time than the Intuitionistic fuzzy Classifier, especially with few mobile nodes. The reason is that the smaller the number of nodes with the same amount of calculation the bigger the time taken.

7.3.2 PKI vs. non-PKI distribution

After the Key size had been determined via the Key size determination function the final problem is to distribute that key among nodes on the network. There were two approaches for the key distribution problem either PKI or non-PKI. In this subsection the results of applying PKI and non-PKI techniques is illustrated as applied in terms of security and processing time

7.3.2.1 Security

The PKI presents more overall security than ordinary non-PKI (single key) that is illustrated by applying both techniques over the network and recording the results regarding to the time required for an external attacker to break the session key.

Table 7.10 and figure 7.10 shows that results under the assumption of using small public-private key pairs

Table 7.10: security of PKI vs, non-PKI

No. nodes	25	50	75	100	125	150	175	200	225	250
Non-PKI	0.15	0.2	0.23	0.26	0.3	0.32	0.36	0.4	0.44	0.45
PKI	0.8	0.85	0.85	0.92	0.93	0.94	0.94	0.94	0.94	0.94

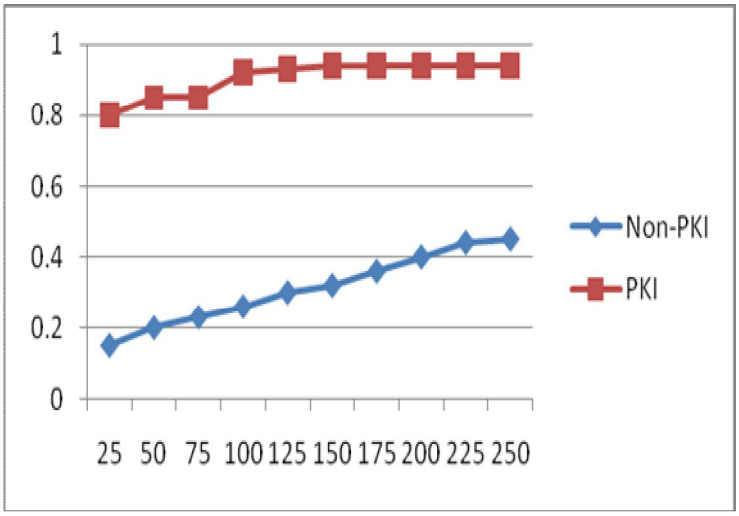


Figure 7.10: security of PKI vs, non-PKI

In graph 7.10 and figure7.10 shows the huge difference in the security level provided by the PKI technique over the Non-PKI mechanism given the same experimental conditions.

7.3.2.2 Processing time

Another factor had been taken into consideration while developing the model that is time required to process the key and distribute it. Table 7.11 and figure 7.11 shows that results under the assumption of using small public-private key pairs

Table 7.11: Processing time of PKI vs. non-PKI

No. nodes	25	50	75	100	125	150	175	200	225	250
Non-PKI	0.32	0.34	0.37	0.37	0.41	0.45	0.47	0.53	0.59	0.60
PKI	0.18	0.30	0.50	0.62	0.68	0.75	0.83	0.87	0.93	0.97

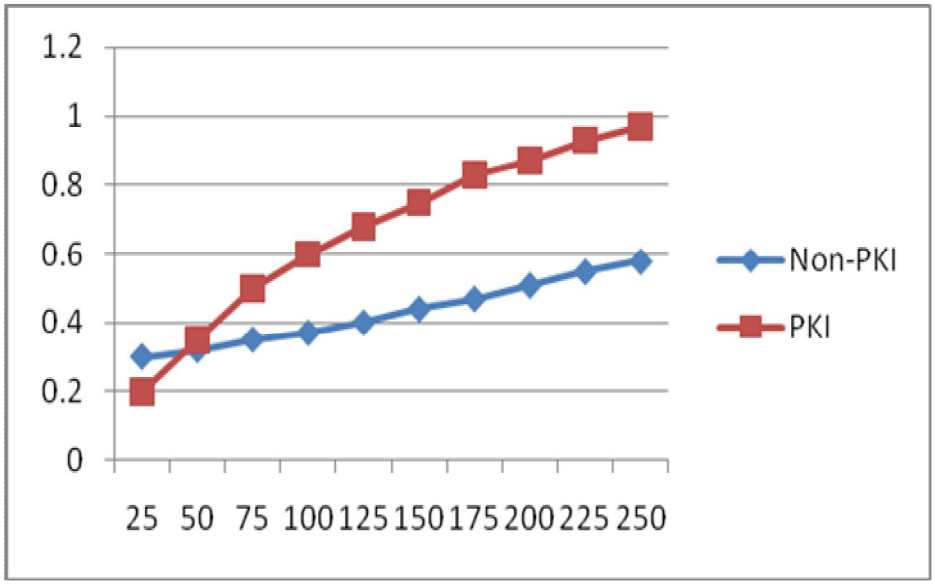


Figure 7.11: Processing time of PKI vs. non-PKI

Table 7.11 and the Figure 7.11 shows that Non-PKI techniques provides relatively small amount of processing time than PKI this due to the amount of modular arithmetic performed in the PKI mechanisms. However the difference in the processing time is neglectable comparing to the security level provided by the PKI under the same conditions.

7.4: Conclusions & Future work

In this section the statement analysis for the studies done through this research is provided as well as the conclusions from the experimental results.

.

In this research the MANET is introduced as one of the important type of networks due to some of its advantages including the disappearance of the regular fixed infra structure and the low cost requirements to operate in such networks.

MANET is subject to various types of attacks and security vulnerabilities in both ways active and passive, so it is urgently required to provide security schemes and mechanisms to stand against these attacks.

The use of PKI as security model to transfer session keys between all nodes in the network shows a very efficient ways of secure transmission in comparison to the ordinary ways or the non-PKI techniques.

AIS is used as a security defence system to stand against the dangerous types of attacks including BOTNETs. Artificial Immune Systems can incorporate many properties of natural immune systems, including diversity, distributed computation, error tolerance, dynamic learning and adaptation and self-monitoring. The human immune system has motivated scientists and engineers for finding powerful information processing algorithms that has solved complex engineering tasks. The Artificial Immune Systems is a general framework for a distributed adaptive system and could, in principle, be applied to many domains. Artificial Immune Systems can be applied to classification problems, optimization tasks and other domains. Like many biologically inspired systems it is adaptive, distributed and autonomous. The primary advantages of the Artificial Immune Systems are that it only

requires positive examples, and the patterns it has learnt can be explicitly examined. In addition, because it is self-organizing, it does not require effort to optimize any system parameters.

Botnet elaborates a huge amount of danger to all types of networks especially MANETs that is because of the diversity nature of MANETs.AIS is found to be one of the most reliable security models that could be used to secure such scenario.

As a future work we recommend to apply the proposed model over different types of networks other than MANETs, we also recommend to create solutions for the problems of trusted third party which poses many challenges to the security designers, another area of problems in MANETs is to find solutions for the problems regarding to authentication using AIS.

References

[1] CR Komala, Srinivas Shetty, S. Padmashree, E. Elevarasi , "Wireless Ad hoc Mobile Networks",National Conference on Computing, Communication and Technology, pp. 168-174, 2010

[2] K. SIVAKUMAR, M.Ph, G. SELVARAJ." OVERVIEW OF VARIOUS ATTACKS IN MANET AND COUNTERMEASURES FOR ATTACKS". International Journal of Computer Science and Management Research Vol 2 Issue 1 January 2013

[3] Md Tanzilur Rahman, Kunal Gupta." MANET: Security Aspects and Challenges" . International Journal of Computer Trends and Technology (IJCTT) – volume 4 Issue 6–June 2013

[4] A. Jaganraj, A. Yogaraj, N. Vignesh, R. V. Anuroop." Handling MANET routing attacks using risk awaremitigation mechanism with distributed node control". Journal Electrical and Electronic Engineering; 1(3): 61-67 ; 2013.

[5] D.S KUTE., A.S. PATIL, N.V PARDAKHE, A.B KATHOLE. "A REVIEW: MANET ROUTING PROTOCOLS AND DIFFERENT TYPES OF ATTACKS IN MANET". International Journal of Wireless Communication. Volume 2, Issue 1 pp.-26-28. 2012

[6] M. Refaei, L. DaSilva, M. Eltoweissy, and T. Nadeem,"Adaptation of Reputation Management Systems to Dynamic Network Conditions in Ad Hoc Networks," IEEE Trans. Computers, vol. 59, no. 5,pp. 707-719, May 2010.

[7] Alex Hinds, Michael Ngulube, Shaoying Zhu, and Hussain Al-Aqrabi." Review of Routing Protocols for Mobile Ad-Hoc NETworks (MANET)". International Journal of Information and Education Technology, Vol. 3, No. 1, February 2013

[8] E. Alotaibi and B. Mukherjee, "A survey on routing algorithms forwireless Ad-Hoc and mesh networks," Computer Networks: TheInternational Journal of Computer and TelecommunicationsNetworking, vol. 56, no. 2, pp. 940–965, October 2011.

[9] X. Hu, J. K. Wang, C. R. Wang, and C. Wang, "Is mobility always harmful to routing protocol performance of MANETs?" in Proc. Of International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery, pp. 108-112, 2010.

[10] Y. Khamayseh, O. M. Darwish, and S. A. Wedian, "MA-AODV: Mobility Aware Routing Protocols for Mobile Ad hoc Networks," in Proc. of Fourth International Conference on Systems and Networks Communications IEEE, pp. 25-29, 2009.

[11] W. Wang and C. Amza, "Motion-based Routing for Opportunistic Ad-hoc Networks," in Proc. of 14th ACM international conference on Modeling, analysis and simulation of wireless and mobile systems, October 31–November 4, pp. 169-178. 2011.

[12] A. Boukerche et al., "Routing protocols in ad hoc networks: A survey," Computer Networks: The International Journal of Computer and Telecommunications Networking, vol. 55, no. 13. pp. 3032–3080, May 2011.

[13] H. Amri, M. Abolhasan, and T. Wysocki, "Scalability of MANET routing protocols for heterogeneous and homogenous networks," Computers and Electrical Engineering, vol. 36, no. 4, pp. 752–765, 2010.

[14] C. Liu and S. Chang, "The study of effectiveness for ad-hoc wireless network," in Proc. of ICIS 2009 2nd International Conference on Interaction Sciences: Information Technology, Culture and Human, Seoul, Korea, 24-26 Nov., 2009, pp. 412-417.

[15] F. Maan and N. Mazhar, "MANET Routing Protocols vs Mobility Models: A Performance Evaluation," in Proc. of Third International Conference on Ubiquitous and Future Networks IEEE, Dalian, China, , pp. 179-184, June 15-17, 2011.

[16] Jaspal Kumar, M. Kulkarni, Daya Gupta." Effect of Black Hole Attack on MANET Routing Protocols". I. J. Computer Network and Information Security, 2013, 5, 64-72

[17] Anuj K. Gupta, Harsh Sadawarti, "Secure Routing Techniques for MANETs", International Journal of Computer Theory and Engineering (IJCTE), ISSN: 1793-8201, Article No. 74, Vol.1 No. 4, pp. – 456-460, October 2009.

[18] Nital Mistry, Devesh. C Jinwala, Mukesh Zaveri, ―Improving AODV Protocol against Black hole Attacks‖, Proceedings of the international multi conference of engineer and computer science vol. 2, 2010

[19] H.A. Esmaili, M.R. Khalili Shoja, Hossein gharaee, "Performance Analysis of AODV under Black Hole Attack through Use of OPNET Simulator", World of Computer Science and Information Technology Journal (WCSIT), Vol. 1, No. 2, 49-52, 2011.

[20] B V Santhosh Krishna, A.L Vallikannu, "Detecting Malicious Nodes For Secure Routing in MANETS Using Reputation Based Mechanism‖", International Journal of Scientific & Engineering Research, Vol. 1, Issue 3, ISSN 2229-5518, December-2010.

[21] Harris Simaremare and Riri Fitri Sari, "Performance Evaluation of AODV variants on DDOS, Blackhole and Malicious Attacks", IJCSNS International Journal of Computer Science and Network Security, VOL.11 No.6, June 2011.

[22] Priyanka Sharma, Yakuta Karkhanawala and K Kotecha, "Bandwidth Constrained Routing of Multimedia Traffic over Hybrid MANETs using Ant Colony Optimization", International Journal of Machine Learning and Computing,2011.

[23] Sergio Cabrero, Xabiel García Pañeda,Thomas Plagemann, David Melendi and Roberto García, "Towards reliable video transmission over sparse MANETs in emergencies", INFORMÁTICA NA EDUCAÇÃO: Teoria & Prática, 2011

[24] Michael Pascoe-Chalke, Javier Gomez, Victor Rangel and Miguel Lopez-Guerrero, "Route duration modeling for mobile ad-hoc networks", Springer, Wireless Networks, 2009

[25] Natarajan Meghanathan, "A Unicast Stable Path Routing Protocol for Mobile Ad hoc Networks based on the Inverse of Link Expiration Time", Network Protocols and Algorithms, ISSN 1943-3581, Vol-3, No-3, 2011.

[26] Priyanka Goyal, Vinti Parmar and Rahul Rishi, "MANET: Vulnerabilities, Challenges, Attacks, Application", International Journal of Computational Engineering & Management, 2011.

[27] M.Renuka and P.Thangaraj, "Reliable Data Security Architecture for Multi-Path Multimedia Streaming in MANET", International Journal of Electronics & Communication Technology, 2012

[28] C. Chen, D. He, S. Chan, J. Bu, Y. Gao and R. Fan, "Lightweight and provably secure user authen-tication with anonymity for the global mobility network",International Journal of Communication Systems, vol. 24, no. 3, pp. 347-362, 2011.

[29] M. S. Bouassida, "Authentication vs. privacy within vehicular ad hoc networks", International Journal of Network Security, vol. 12, no. 3, pp. 256-272, 2011.

[30] U. Aickelin, , P.Bentley, , S.Cayzer, J. Kim, and J. McLeod, , "Danger Theory: The Link between AIS and IDS?", Hewlett Packard Labs: HPL-2003-138, 16 July, 2003.

[31] D.Dasgupta, and S.Yu, , “Artificial Immune Systems: A ibliography,” CS Technical Report: CS-03-002, Computer Science Division, The University of Memphis, USA, Dec, 2003.

[32] L. N.de Castro, and J. Timmis, "Artificial Immune Systems: A New Combutational Intelligence Approach", Springer-Verlag, 2002.

[33] D.Dasgupta, Z. Ji, and F.Gonzalez, , "Artificial Immune System (AIS) Research in the Last Five Years", Proc. Canberra, Australia: IEEE Press, 8 pp. 123–130. 2003.

[34] M.Cohn, , “An alternative to current thinking about positive selection, negative selection and activation of T cells,” Immunology, Blackwell Publishing, vol. 111,, pp. 375–380., 2004.

[35] C. Janeway, and P.Travers, , "Immunobiology : the immune system in health and disease, Current Biology"; London, San Francisco New York: Garland Publishers, 5th Edition, 2001.

[36] G. Gu, R. Perdisci, J. Zhang, and W. Lee. Botminer." Clustering analysis of network traffic for protocol- and structure-independent Botnet detection". In Security , 2008.

[37] S.R . Jathe,. and D.M. Dakhane, "A Review Paper on Black Hole Attack and Comparison of Different Back Hole Attack Techniques". International Journal of Cryptography and Security, 2(1): p. 22-26,2012

[38] A. Nummipuro, “Detecting P2P-Controlled Bots on the Host”, Seminar on Network Security, Espoo, Helsinki, 2007.

[39] U.Venkanna, and R.L. Velusamy. "Black hole attack and their counter measure based on trust management in MANET": A survey. 2011.

[40] S. Chang and T. E. Daniels. "correlation based node behavior profiling for enterprise network secure"ty. In SECURWARE, 2009.

[41] Paul K. Harmer, Paul D. Williams, Gregg H. Gunsch, and Gary B. Lamont. "An Artificial Immune System Architecture for Computer Security Applications". IEEE TRANSACTIONS ON Maziar Janbeglou, Mazdak Zamani, Suhaimi Ibrahim. Improving the Security of Protected Wireless Internet Access from Insider Attacks. Advances in information Sciences and Service Sciences. Volume4, Number12, July 2012.

[42] M. Abolhasan, , T. Wysocki, and E. Dutkiewicz, "A review of routing protocols for mobile ad hoc networks". Ad Hoc Networks,. 2(1): p. 1-22., 2004

[43] Mojtaba Alizadeh, Mazdak Zamani, Ali Rafiei Shahemabadi, Jafar Shayan, Ahmad Azarnik. "A Survey on Attacks in RFID Networks". Open International Journal of Informatics (OIJI). Vol 1 (2012).

[44] S. Dokurer,, "Simulation of Black hole attack in wireless Ad-hoc networks". Atılım University.2006.

[45] M. Malik, "On Demand Routing Protocols in Mobile Networks. " International Journal of Research in Science And Technology (ijrst). 1(Apr-Jun). 2012

[46] K. Reddy. and P. Thilagam, Taxonomy of Network Layer Attacks in Wireless Mesh Network. Advances in Computer Science, Engineering & Applications, 2012: p. 927-935.

[47] S. Djahel, , F. Naït-Abdesselam, and Z. Zhang, "Mitigating packet dropping problem in mobile ad hoc networks": Proposals and hallenges. IEEE Communications Surveys and Tutorials. 13(4): p. 658-672.2011

[48] C. Lin, , "AODV routing implementation for scalable wireless Ad-hoc network simulation (SWANS)". httpy/jist. ece. cornell. 2004.

[49] L Tamilselvan,. and V. Sankaranarayanan, "Prevention of co-operative black hole attack in MANET". Journal of Networks, 2008. 3(5): p. 13-20.

[50] J.H Song,., V.W.S. Wong, and V. Leung. "A framework of secure location service for position-based ad hoc routing". in Proceedings of the 1st ACM international workshop onPerformance evaluation of wireless ad hoc, sensor, and ubiquitous networks. 200

[51] P.N. Raj, and P.B. Swadas, DPRAODV."A Dyanamic Learning System Against Blackhole Attack In Aodv Based Manet. 2009.

[52] Juki Wirawan Tantra, Chuan Heng Foh and Dongyu Qiu, “On Link Reliability in Wireless Mobile Ad Hoc Networks”, IEEE 64th Vehicular Technology Conference, pp-1-5, 2006.

[53] P.C. Tsou, "Developing a BDSR scheme to avoid black hole attack based on proactive and reactive architecture in MANETs." 2011:

[54] Saeed Yazdanpanah, Saman Shojae Chaeikar, Mazdak Zamani and Reza Kourdi. "Security Features Comparison of Master Key and Ikm Cryptographic Key Management for Researchers and Developers". 3rd International Conference on Software Technology and Engineering. Kuala Lumpur, Malaysia August 12-13, 2011.

[55] Maryam Gharooni, Mazdak Zamani, and Mehdi Mansourizadeh. "A Confidential RFID Model to Prevent Unauthorized Access". 3rd International Conference on Information Science and Engineering. Yangzhou, China. Sep 29- Oct 1, 2011

[56] Mojtaba Ali Zadeh, Mazleena Salleh, Mazdak Zamani, Jafar Shayan, Sasan Karamizadeh. "Security and Performance Evaluation of Lightweight Cryptographic Algorithms in RFID". 16th WSEAS International Conference on Communications. Kos Island, Greece. July 14-17, 2012.

[57] J. R. Binkley, and S. Singh, "An Algorithm for Anomaly-based Detection", Proceedings of USENIX Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI), , pp. 43–48 July 2006

[58] Eghbal Ghazizadeh, Mazdak Zamani, Jamalullail Ab Manan,Reza Khaleghparast, Ali Taherian. A Trust Based Model for Federated Identity Architecture to Mitigate Identity Theft. The 7th International Conference for Internet Technology and Secured Transactions. London, UK. 10th- 12th December 2012.

[59] A.Lavanya, K.Saravanan, M.E Scholar." A REVIEW OF DDoS ATTACKS IN MOBILE AD-HOC NETWORKS", International Journal of Societal Applications of Computer Science Vol 1 Issue 1 November 2012

[60] S. Nishanthi ." Intrusion Detection in Wireless Sensor Networks Using Watchdog Based Clonal Selection Algorithm ". IJREAT International Journal of Research in Engineering & Advanced Technology, Volume 1, Issue 1, March, 2013

[61] Esponda, F., Forrest, S. and Helman, P., "A Formal Framework for Positive and Negative Detection Schemes," IEEE Transactions on Systems, Man, and Cybernetics- Part B: Cybernetics, vol. 34, no. 1, 2004, pp. 357–373.

[62] Eghbal Ghazizadeh, Mazdak Zamani, Jamalul- Lail Ab Manan and Abolghasem Pashang. A Survey on Security Issues of Federated Identity in the Cloud Computing. The 4th IEEE International Conference on Cloud Computing Technology and Science (CloudCom 2011). Dec 3 – 6, 2012. Taipei, Taiwan.

[63] A. Ramachandran, N. Feamster, and D. Dagon, "Revealing Botnet membership using DNSBL counter-intelligence," in Proceedings of the 2nd Conference on Steps to Reducing Unwanted Traffic on the Internet Workshop (SRUTI '06), vol. 2, p. 8, San Jose, Calif, USA, 2006.

[64] P. Barford and V. Yegneswaran, "An inside look at Botnets," in Proceedings of the ARO-DHS Special Workshop on Malware Detection, Advances in Information Security, Springer, 2006..

[65] W. T. Strayer, R.Walsh, C. Livadas, and D. Lapsley, "Detecting Botnets with tight command and control," in Proceedings of the 31st Annual IEEE Conference on Local Computer Networks (LCN '06), pp. 195–202, Tampa, Fla, USA, November 2006.

[66] M. Akiyama, T. Kawamoto, M. Shimamura, T. Yokoyama, Y. Kadobayashi, and S. Yamaguchi, "A proposal of metrics for Botnet detection based on its cooperative behavior," in Proceedings of the International Symposium on Applications and the Internet Workshops, p. 82, Washington, DC, USA, January2007

[67] Shohreh Honarbakhsh, Mazdak Zamani, Roza Honarbakhsh. Dynamic Monitoring in Ad hoc Network. Applied Mechanics and Materials. Vols. 229-231. pp 1481-1486. (2012) Trans Tech Publications, Switzerland.

[68] Bhatia, R., D. Gupta, and S.K. Vijay, Security Issues Prtaining to AD-HOC Networks-A Survey. IJCSMS International Journal of Computer Science and Management Studies, 2012. 12(01, January 2012).

[69] K.Gorantala, ," Routing protocols in mobile adhoc networks". Master's Thesis in Computing Science, June, 2006. 15.

[70] Hossein Rouhani Zeidanloo, Azizah Abdul Manaf, Rabiah Bt Ahmad, Mazdak Zamani and Saman Shojae Chaeikar. A Proposed Framework for P2P Botnet Detection. IACSIT International Journal of Engineering and Technology (IJET), Vol.2, No.2, April 2010, ISSN 1793-8236.

[71] Geetika, Naveen Kumari." Detection and Prevention Algorithms of DDOS Attack in MANETs". International Journal of Advanced Research in Computer Science and Software Engineering. Volume 3, Issue 8, August 2013 ISSN: 2277 128X.

[72] K. Mohan Mali, A. Pramod Jadhav, " Review of DDoS and Flooding Attacks in MANET", International Journal of Emerging Technology and Advanced Engineering, Volume 3, Issue 5, May 2013.

[73] K. Sivakumar, Dr. G. Selvaraj,"Overview of various Attack in MANET and Countermeasures for Attack", International Journal of Computer Science and Management Research, Vol 2 Issue 1, January 2013.

[74] J. Govil, "Examining the criminology of bot zoo," in Proceedings of the 6th International Conference on Information, Communications and Signal Processing (ICICS '07), pp. 1–6, Singapore, December 2007.

[75] Laxmi Bala, A.K. Vatsa, "Quality based Bottom-up-Detection and Prevention Techniques for DDOS in MANET",International Journal of Computer Applications, Volume 55– No.2, October 2012.

[76] Minda Xiang,Yu Chen,Wei-Shinn Ku, Zhou Su, " Mitigating DDoS Attacks using Protection Nodes in Mobile Ad Hoc Networks", Dept. of Computer Science & Software Engineering, Auburn University, Auburn, AL 36849.

[77] R. Puri, "Bots and Botnets: an overview," Tech. Rep., SANS Institute, 2003

[78] Priyanka Goyal, Vinti Parmar, Rahul Rishi, "Vulnerabilities, Challenges, Attacks, Application",International Journal of Computational Engineering & Management, Vol. 11, January 2011.

[79] Prajeet Sharma,Niresh Sharma,Rajdeep Singh," A Secure Intrusion detection system against DDOS attack in Wireless Mobile Ad-hoc Network", International Journal of Computer Applications,Volume 41– No.21, March 2012.

[80] Rizwan Khan , A. K. Vatsa, "Detection and Control of DDOS Attacks over Reputation and Score Based MANET", Journal of Emerging Trends in Computing and Information Sciences, VOL. 2, NO. 11, October 2011.

[81] David Dittrich, Sven Dietrich." Discovery techniques for P2P Botnets" Stevens CS Technical Report 2008-4, September 2008 Last revision: 21 April 2009.

[82] J. Grizzard, V. Sharma, C. Nunnery, B. Kang and D. Dagon, "Peerto- Peer Botnets: Overview and Case Study", In HotBots '07 conference, Usenix, 2007.

[83] M. STEGGINK and I. IDZIEJCZAK, "Detection of peer-to-peer Botnets", University of Amsterdam, Netherlands, 2007

[84] T. Holz, M. Steiner, F. Dahl, E.W. Biersack and F. Freiling, "Measurements and Mitigation of Peer-to-Peer-based Botnets: A Case Study on Storm Worm", 1st USENIX Workshop on Large-Scale Exploits and Emergent Threats, Usenix, San Francisco, 2008.

[85] F P. Porras, H. Saidi and V. Yegneswaran, "A Multi-perspective Analysis of the Storm (Peacomm)Worm", Computer Science Laboratory, SRI International , CA, 2007.

[86] D. Dittrich and S. Dietrich. Discovery techniques for P2P botnets. Technical Report CS 2008-4, Stevens Institute of Technology, September 2008.

[87] D. Dittrich and S. Dietrich. New directions in P2P malware. In Proceedings of the 2008 IEEE Sarnoff Symposium, page 5, Princeton, New Jersey, USA, April 2008.

[88] D. Dittrich and S. Dietrich. Technical Report CS 2008-3 Stevens Institute of Technology, June 2008.

[89] C. A. J. Bambenek. Botnets: Proactive system defense. In RO-DARPA-DHS Special Workshop on Botnets, 2006.

[90] S. Chang, L. Zhang, Y. Guan, and T. Daniels. A framework for p2p botnets. In International Conference on Communications and Mobile Computing, Jan, 2009.

[91] M. Collins, T. Shimeall, S. Faber, J. Janies, R.Weaver, M. D. Shon, and J. Kadane. Using uncleanliness to predict future botnet addresses. In IMC, 2007.

[92] D. Dittrich and S. Dietrich. P2p as botnet command and control: a deeper insight. In Malware, 2008.

[93] J. Goebel and T. H. Rishi. Identify bot contaminated hosts by irc nickname evaluation. In HotBots, 2007.

[94] D. Dittrich and S. Dietrich. Discovery techniques for P2P Botnets. Technical Report CS 2008-4, Stevens Institute of Technology, September 2008.

[95] D. Dittrich and S. Dietrich. New directions in P2P malware. In Proceedings of the 2008 IEEE Sarnoff Symposium, page 5, Princeton, New Jersey, USA, April 2008.

[96] S. Hofmeyr, and S. Forrest,, "Architecture for an Artificial Immune System," Evolutionary Computation, vol. 7, no. 1, 2000, pp. 1289– 296.

[97] M. Cohn,, "Tritope model of restrictive recognition by the TCR," Trends in Immunology, vol. 24, no. 3, 2003, pp. 127–131.

[98] Somayaji, A. and Forrest, S., "Automated Response Using System-Call Delays," Proc. Usenix, San Diego, California, 2000.

[99] Hofmeyr, S. and Forrest, S., "Immunity by Design: An Artificial Immune System," Proc. Genetic and Evolutionary Computation Conference (GECCO), Edited by. Banzhaf, W. et. al., pp. 1289–1296 2006,.

[100] P.Williams, , K. Anchor, , J. Bebo,., G. Lamont,. and G. Gunsch, "CDIS: Towards a Computer Immune System for Detecting Network Intrusions," Lecture Notes in Computer Science, vol. 2212, , pp. 117–133. 2005

[101] J.Kim, , A.Ong, and R. E. Overill, , "Design of an Artificial Immune System as a Novel Anomaly Detector for Combating Financial Fraud in the Retail Sector," Proc. Congress on Evolutionary Computation, Edited by. Sarker, R. et. al., Canberra, Australia: IEEE Press, 405–412.2003.

[102] De Castro LN, Von Zuben FJ An evolutionary immune network for data clustering. In: Proceedings of the IEEE Brazilian symposium on neural networks, pp 84–89, 2006

[103] K.Seshadri Ramana et al. " Artificial Immune System New Paradigm "/ (IJCSE) International Journal on Computer Science and Engineering Vol. 02, No. 02, 259-263, 2010

[104] X. Wei and L. Zhi, "fie multi-objective routing optimization of WSNs based on an improved ant colony algorithm," in Proceedings of the 6th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM '10), September 2010.

[105] J. Dollner and K. Hinrichs, "A generic rendering system," IEEE Transactions of Visualization and Computer Graphics, vol. 8, no. 2, pp. 99–118, 2002.

[106] H. He, Z. Zhu, and E. Mäkinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," IEEE Transactions on Neural Networks, vol. 20, no. 6, pp. 973–982, 2009.

[107] C. M. Ionescu, I. Muntean, J. A. Tenreiro-Machado, R. De Keyser, and M. Abrudean, "A theoretical study on modeling the respiratory tract with ladder networks by means of intrinsic fractal geometry," IEEE Transactions on Bio-Medical Engineering, vol. 57, no. 2, pp. 246–253, 2010.

[108] J. Podpora, L. Reznik, and G. Von Pless, "Intelligent realtime adaptation for power efficiency in sensor networks," IEEE Sensors Journal, vol. 8, no. 12, pp. 2066–2073, 2008.

[109] A. I. Moustapha and R. R. Selmic, "Wireless sensor network modeling using modi□ed recurrent neural networks: application to fault detection," IEEE Transactions on Instrumentation and Measurement, vol. 57, no. 5, pp. 981–988, 2008.

[110] H. He, Z. Zhu, and E. Mäkinen, "A neural network model to minimize the connected dominating set for self-configuration of wireless sensor networks," IEEE Transactions on Neural Networks, vol. 20, no. 6, pp. 973–982, 2009.

[111] S. Sar, janov and J. Y. Le Boudec, "An artificial immune system approach with secondary response for misbehavior detection in mobile ad hoc network," IEEE Transactions on Neural Networks, vol. 16, no. 5, pp. 1076–1087, 2005.

[112] H. Wang, D. Peng, W. Wang et al., "Artificial immune system based image pattern recognition in energy efficient wireless multimedia sensor networks," in Proceedings of the IEEE Military Communications Conference (MILCOM '08), November 2008.

[113] S. Yang, H. Cheng, and F. Wang, "Genetic algorithms with immigrants and memory schemes for dynamic shortest path routing problems in mobile ad hoc networks," IEEE Transactions on Systems, Man and Cybernetics Part C, vol. 40, no. 1, pp. 52–63, 2010.

[114] T. Morrison and U. Aickelin" An Artificial Immune Systems as a Recommender System for Web Sites", in Proceedings of the 1st International Conference on Artificial Immune Systems (ICARIS-2002), pp 161-169, Canterbury, UK.,2004

[115] J. A.,Roubos, M.Setnes J.Abonyi "Learning fuzzy classification rules from labeled data. Information Science 150, pp. 77–93 ,2003.

[116] R. T. Alves, , et al." An artificial immune system for fuzzy-rule induction in data mining". Parallel Problem Solving from Nature—PPSN VIII. LNCS, vol. 3242, pp. 1011–1020. Springer Heidelberg 2004.

[117] D.Dasgupta,, "Artificial Immune Systems and Their Applications.", Spring-Verlag Berlin Heidelberg Germany 2007.

[118] F. A.Gonzales, , D Dasgupta,." An Immunogenetic Technique to Detect Anomalies in Network Traffic". In: Proceedings of Genetic and Evolutionary Computation. pp. 1081–1088. Morgan Kaufmann San Mateo (2002).

[119] C. Marsala." Fuzzy Partitioning Methods, Granular Computing: An Emerging Paradigm". Physica-Verlag GmbH Heidelberg Germany, pp. 163–186 .2001

[120] O. Nasaroui., F. Gonzales., D. Dasgupta.: The Fuzzy Artificial Immune System: Motivations, Basic Concepts, and Application to Clustering and Web Profiling. In: Proceedings of IEEE International Conference on Fuzzy Systems, pp. 711–716 ,2002..

[121] P. Bentley and J. Timmis." A Fractal Immune Network". Proceeding of the Third Conference ICARIS, pages 133-145, Edinburg, UK, September. Springer. 2005

[122] X.Bian and J. Qiu.." Adaptive Clonal Algorithm and Its Application for Optimal PMU Placement". Proceedings of 2006 International Conference on Communications, Circuits and Systems, 25-28 June, Volume: 3, on page(s): 2102-2106.2006.

[123] F. Campelo., F. Guimaraes.,H. Igarashi and J. Ramirez ,. "A Clonal Selection Algorithm for Optimization in Electromagnetics". IEEE Transactions on Magnetics, VOL. 41, NO. 5. 2005

[124] L. Castro de and J. Timmis ,. “An Artificial Immune Network for Multimodal Function Optimization”. Proceedings of IEEE Congress on Evolutionary Computation (CEC'02), vol. 1, pp. 699-674, May, 2002 Hawaii

[125] L. Castro. de and J. Timmis.” Artificial Immune Systems as a Novel Soft Computing Paradigm”. Soft Computing Journal, vol. 7, Issue 7. 2003

[126] J. Y. Le Boudec and S. Sara_janovic. An Artificial Immune System Approach to Misbehavior Detection in Mobile Ad-Hoc Networks. Proceedings of Bio-ADIT. , pp. 96-111. January 2004 Lausanne, Switzerland

[127] S. Sara_janovic and J. Y. Le Boudec. “An Artificial Immune System Approach with Secondary Response for Misbehavior Detection in Mobile Ad-Hoc Networks”. TechReport IC/2003/65, EPFL-DI-ICA, Lausanne, Switzerland, November 2003.

[128] J. Jung and E. Sit, “An empirical study of spam traffic and the use of DNS black lists,” in Proceedings of the 4th ACM SIGCOMM Internet Measurement Conference (IMC ’04), pp. 370–378, Taormina, Italy, October 2007.

[129] Fu J., Li Z., and Tan H.,.” A Hybrid Artificial Immune Network with Swarm Learning”. 2007 International Conference on Communications, Circuits and Systems (ICCCAS’07), 11-13 July, Kokura, On page(s): 910-914. 2007

[130] Z. Gan, G. Li, Z. Yang., and M.Jiang ,.” Automatic Modeling of Complex Functions with Clonal Selection-based Gene Expression Programming. Third International Conference on Natural Computation (ICNC 2007), Haikou , 24-27 Aug., Volume: 4, On page(s): 228-232. 2007