



A Threshold Secret Sharing Scheme for Self- Securing

Noisy Mobile Ad Hoc Networks

إسلوب المشاركة الأمنية (TSS) للحماية الذاتية في الشبكات اللاسلكية العشوائية
المتنقلة المشوشة

By

Tamara J. Ishak Stephan

(20043240528)

Supervisors

Dr.Hilal Al-Bayatti and Dr. Hussein Al-Bahadili

This thesis is submitted to the Department of Computer Science, Graduate
College of Computing Studies, Amman Arab University for Graduate
Studies in partial fulfillment of the requirement for the degree of Master of
Science (M.Sc) in Computer Science.

**Graduate College of Computing Studies
Amman Arab University for Graduate Studies
(February -2009)**

Authorization of Dissemination

I the undersigned "Tamara J. Ishak Stephan" authorize hereby Amman Arab University of Graduate Studies to provide copies of this thesis to libraries, institutions and any other parties upon their request.

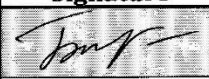


Name: Tamara J. Ishak Stephan

Signature: 

Date: 20/5/2009

Resolution of the Examining committee

This dissertation titled “A threshold Secret Sharing Scheme for Self-Securing Noisy Mobile Ad Hoc Networks”, has been defended and approved on 20/5/2009.

Examining Committee	Title	Signature
Dr. Basil Kasasbeh	Chair	
Dr. Hussein Al-Bahadili	Member and Supervisor	
Dr. Mezher Al-Ani	Member	

Acknowledgment

First and foremost, I would like to express my sincere gratitude and appreciation to my supervisors Prof. Hilal Al-bayatti Dr. Hussein Al-Bahadili for their guidance during all stages of this research, for answering endless questions, for their great support, professional advice, and profound understanding. I would especially like to thank them for their extensive comments on the thesis write up, as they have also helped me to present my thesis in a much better way. Always their encouragement was a great source of motivation for me.

I also would like to thank all members of staff at Amman Arab University for Graduate Studies, in particular, the members of staff at the Graduate College of Computing Studies and special thanks to the dean Prof. Alaa Al-Hamami.

Finally, it would be unthinkable of me not to thank my wonderful family and specially my sister Dr. Jane J. Ishak Stephan for always encouraging me to pursue my beliefs, my ideas and my dreams. In addition, I thank them for their unwavering support, motivation and invaluable help in finishing this thesis.

Table of Contents

Acknowledgment.....	iv
TABLE OF CONTENTS	V
LIST OF FIGURES	VIII
LIST OF TABLES.....	IX
LIST OF ABBREVIATIONS	X
ABSTRACT	XI
CHAPTER 1 INTRODUCTION	1
1.1. MOBILE AD HOC NETWORKS (MANETs)	1
1.1.1 DEFINITION OF MANET	1
1.1.2 APPLICATIONS OF MANETs	3
1.1.3 CHALLENGES AND LIMITATIONS TO MANETs.....	4
1.2.SECURITY IN MANETs	4
1.2.1 REQUIREMENTS OF MANETs SECURITY.....	4
1.2.2 CHALLENGES AND LIMITATIONS TO MANETs SECURITY.....	5
1.3. AUTHENTICATION.....	6
1.3.1.DEFINITIONS.....	6
1.3.2 AUTHENTICATION TECHNIQUES	8
Public-key cryptography	10
1.4.WIRELESS NETWORK ENVIRONMENTS.....	12
1.4.1 NOISELESS (ERROR-FREE) ENVIRONMENT	12

1.4.2 NOISY (ERROR-PRUNE) ENVIRONMENT	12
1.5.PROBLEM STATEMENT	13
1.6. RESEARCH OBJECTIVES.....	14
1.7.THESIS ORGANIZATION	14
CHAPTER 2 LITERATURE REVIEWS	16
CHAPTER 3 THE THRESHOLD SECRET SHARING (TSS) SCHEME.....	28
3.1 .RSA ASYMMETRIC PUBLIC-KEY SECURITY ALGORITHM	29
3.2 .AUTHENTICATION MODELS	32
3.2.1 THE TTP MODEL	33
3.2.2 THE LOCALIZED-TRUST MODEL	34
3.3.CONCEPT OF SHAMIR’S SECRET SHARING SCHEME	35
3.4. THE PROPOSED TSS SCHEME	37
3.4.1 CONCEPT OF THE PROPOSED TSS SCHEME.....	37
3.4.2 IMPLEMENTATION	38
3.4.3 THE PROPOSED TSS SCHEME LOCALIZED CERTIFICATION PROCEDURE	40
3.4.4 TRUSTIBILITY FACTOR (T_F).....	41
3.5. SYSTEM, ADVERSARY, AND INTRUSION MODELS	42
3.5.1 SYSTEM MODEL.....	42
3.5.2 ADVERSARY MODELS	43
3.5.3 INTRUSION MODEL.....	44
3.6. SIMULATION MODEL.....	45
3.6.1 NETWORK SIMULATION.....	45
3.6.2 THE NETWORK SIMULATOR (MANSIM).....	46

3.7.PERFORMANCE MEASURES	48
3.8 .PRACTICAL IMPLEMENTATION OF THE PROPOSED TSS SCHEME.....	50
CHAPTER 4 SIMULATION RESULTS AND DISCUSSIONS	52
4.1. SCENARIO #1: INVESTIGATE THE EFFECT OF NODE DENSITY (N)	52
4.2.SCENARIO #2: INVESTIGATE THE EFFECT OF NODE MOBILITY (U).....	57
4.3. SCENARIO #3: INVESTIGATE THE EFFECT OF NODE RADIO TRANSMISSION RANGE (R).....	61
4.4 SCENARIO #4: INVESTIGATE THE EFFECT OF PROBABILITY OF RECEPTION (PC)	64
CHAPTER 5 CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK	69
5.1.CONCLUSIONS.....	69
5.2.RECOMMENDATIONS FOR FUTURE WORK	70
REFERENCES	71
ARABIC SUMMARY	79

List of Figures

<u>Figure</u>	<u>Title</u>	<u>Page</u>
1.1	An infrastructure (access point) network_	12
1.2	An infrastructureless (ad hoc) networks	12
3.1	Localized certification procedure of the proposed TSS scheme	47
3.2	An algorithm for handling misbehaving nodes	49
3.3	Computational module of the TSS algorithm	55
4.1	Variation of S_R with k for various values of n and p_c	60
4.2	Sensitivity of $S_R (S(k))$ for Scenario #1	62
4.3	Variation of S_R with k for various u and p_c	64
4.4	Node distribution at time t and $t+\Delta$	65
4.5	Variation of S_R with k for various values of R and p_c	68
4.6	Sensitivity of $S_R (S(k))$ for Scenario #3	69
4.7	Variation of S_R with k for various values of p_c .	71
4.8	Sensitivity of $S_R (S(k))$ for Scenario #4	72

List of Tables

<u>Tables</u>	<u>Title</u>	<u>Page</u>
4.1	Input parameters for Scenario #1	59
4.2	Variations of S_R with k for various values of n	60
4.3	Variations of $S(k)$ with k for various values of n	61
4.4	Input parameters for Scenario #2	63
4.5	Variations of S_R with k for various values of u	64
4.6	Input parameters for Scenario #3	66
4.7	Variations of S_R with k for various values of R	67
4.8	Variations of $S(k)$ with k for various values of R	68
4.9	Input parameters for Scenario #4	70
4.10	Variations of S_R with k for various values of p_c	70
4.11	Variations of $S(k)$ with k for various values of p_c	72

List of Abbreviations

AP	Access Point
BS	Base Station
CA	Certificate Authority
CASK	Certificate Authority Signing Key
DoS	Denial-of-Service
DS	Digital Signature
KDC	Key Distribution Center
MANET	Mobile Ad-hoc Network
MAC	Message Authentication Code
PDA's	Personal Digital Adapters
PGP	Pretty Good Privacy
KU	Public Key
QoS	Quality-of-Service
RSA	Rivest–Shamir-Adelman
KR	Private Key
TTP	Trusted Third Party
TSS	Threshold Secret Sharing
TTS	Threshold Trust Security
WAP	Wireless access point
WLAN	Wireless Local Area Network

Abstract

The threshold secret sharing (TSS) scheme proposed by A. Shamir has been widely used to provide distributed authentication services for self-securing wireless ad hoc networks. Many researches have been carried-out to investigate the performance of this scheme in noiseless (error-free) wireless ad hoc networks, where it demonstrated an excellent performance in terms of providing a high authentication (certification) success ratio, reliability, scalability, minimum convergence time, and reasonable communications overhead and delay. However, in practice, wireless networks suffer from high packet-loss due to the presence of noise and node mobility, which may significantly affect the performance of this scheme.

The main objective of this work is to develop and evaluate the performance of an authentication scheme for self-securing mobile ad hoc networks (MANETs) suffering from high packet-loss (i.e., noisy MANETs) and node mobility. The scheme is based on Shamir's TSS concept, and therefore, it is referred to as the TSS scheme. It was implemented using the platform of the MANET networks simulator (MANSim). MANSim is a discrete-event process-oriented research-level network simulator developed using C++. The main feature of our implementation is that the authentication can be performed in every network neighborhood; this feature is so important to authenticate roaming users in a MANET. In addition, the network is not exposed to any single point of compromise, single point of denial-of-service (DoS) attack, or single point of failure.

In order to evaluate the performance of the TSS scheme, a number of scenarios were simulated. These scenarios illustrate the variation of the success ratio, which is defined as the number of successful authentication services over all requests during a certain simulation time, with the threshold secret shares for various node densities, node speeds, node radio transmission ranges, network noise-level (expressed in terms of probability of receptions). The outcomes of these scenarios are so important to facilitate efficient network management. According to the

results obtained, we concluded that presence of noise inflicts significant reduction in the success ratio and consequently degrades the performance of the network, while node mobility inflicts no or insignificant effects.

CHAPTER 1

INTRODUCTION

1.1. Mobile Ad Hoc Networks (MANETs)

1.1.1 Definition of MANET

Wireless networks usually consist of a number of communication devices (e.g., computers, microprocessor-based devices, personal digital adapters (PDAs), mobile phones, and/or any digital devices with compatible communication capabilities) that are connected without using wires. Instead they are utilizing radio waves to enable communication between devices in a limited coverage area. This allows communication devices (also called nodes) to move around within the broad radio coverage area and still be connected to the network [Mur 04, Per 00].

Wireless networks that are using the IEEE 802.11 wireless local area network (WLAN) protocols can be configured to operate in one of the following networks configurations [Tan 03]:

- (1) Access point (AP) network: In which nodes communicate with each other through a base station (BS) or AP that works as a centralized controller; therefore, it is referred to as an infrastructure network as shown in Figure (1.1).
- (2) Ad hoc networks: In which nodes communicate with each other directly without relying on any infrastructure or centralized controller; therefore, it is referred to as an infrastructureless network as shown in Figure (1.2).

In the first configuration, a wireless access point (WAP), which is a device that connects wireless devices together to form a wireless network, is used as a centralized controller as shown in Figure (1.1). In addition, a WAP can be used to connect wire and wireless networks together and relay data between these different networks. Due to the nature of the radio links, nodes are allowed to be mobile within the WAP coverage area.

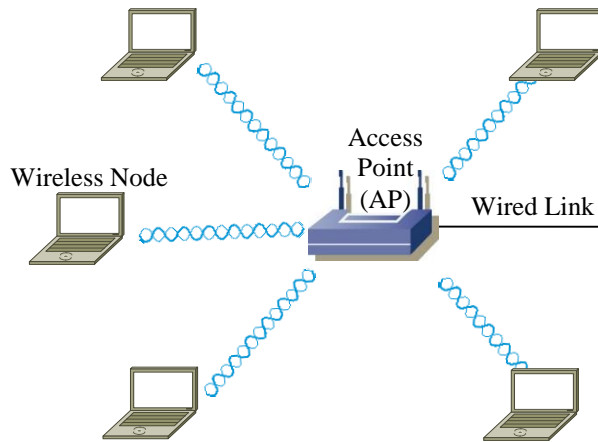


Figure (1.1) – An infrastructure (access point) network.

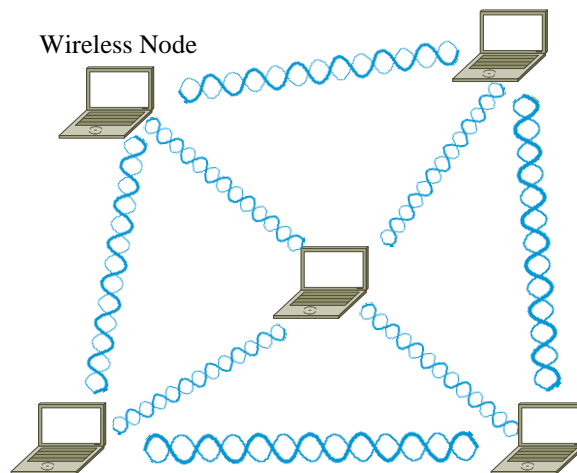


Figure (1.2) – An infrastructureless (ad hoc) networks.

Furthermore, several WAPs can be linked together to form a larger network, similar to cellular mobile phone networks [Hei 99], that allows the exchange of data between devices connected to different BSs. As the node of one WAP travels into the range of another, a "hand off" occurs from the old WAP to the new one and the

node is able to continue communication seamlessly throughout the network. Typical applications of this type of network include office and campus WLANs [Mur 04].

In contrast to the centralized control WAP networks are the ad hoc wireless networks, in which nodes manage themselves without the need for any WAP or centralized controller as shown in Figure (1.2). Once again, due to the nature of the radio links that connect nodes, nodes are allowed to move around and retain its connectivity to the network, therefore, such networks are called MANETs [Agr 03].

In MANETs, due to the limited radio range of the mobile nodes, it may be necessary for one mobile node to enlist the aid of other nodes in forwarding data packets to their destination. Thus, each mobile node operates not only as a host but also as a router using a specific routing mechanism (routing protocol) to forward data packets, efficiently and reliably, for other mobile nodes within the network, which may not be within the radio transmission range of the source.

1.1.2 Applications of MANETs

There has been a tremendous growth in the use of MANETs, not only due the development in the technology but also due to their high flexibility. MANETs can be used wherever there is a prompt need for establishing a networking environment for a limited duration of time. These networks provide cost-effective tremendous opportunities and can be used in numerous situations, particularly, where a communication infrastructure is non-existent or difficult to establish within timing constraints. They also provide an alternative infrastructure in case of failure of the conventional one, as after a disaster and more.

Typical applications of MANETs may include [Sun 01]:

- Industrial environment applications.
- Academic environment applications.

- Healthcare applications.
- Military battlefield.
- Search and rescue operations.

1.1.3 Challenges and limitations to MANETs

While MANETs offer benefits over wired and other wireless networks, there are many challenges and limitations that need to be addressed for fully harvesting MANETs benefits. These limitations arise from the physical properties of the transmission medium in which the MANETs operate, and the limitations forced by mobility which causes dynamically changing topologies and routes [Sta 03].

In general, the main challenges and limitations to the use of MANETs that need to be carefully considered are: limited communication bandwidth and capacity, limited battery power and lifetime, size of the mobile devices, information security, communication overhead, induced transmission errors, distributed control problem, nodes mobility and dynamic variation of network topology, scalability, meeting certain Quality-of-Service (QoS), etc [For 07, Tan 03, Per 00].

1.2.Security in MANETs

1.2.1 Requirements of MANETs Security

The main requirements that need to be carefully considered to ensure high-level of MANETs security are [For 08, Cap 03a, Sta 03]:

- (1) Confidentiality: The data sent by the sender (source node) must be comprehensible only to the intended receiver (destination node). Though an intruder might get hold of the data being sent, it must not be able to derive or very hard to derive any useful information out of the data. One of the popular techniques used for ensuring confidentiality is data encryption.

- (2) **Authentication:** Enables a node to ensure the identity of the peer node it is communicating with. Without authentication, an adversary could masquerade a node, thus gaining unauthorized access to resource and sensitive information and interfering with the operation of other nodes.
- (3) **Integrity:** The data sent by the source node should reach the destination node as it was sent: unaltered never corrupted. A data could be corrupted because of benign failures, such as radio propagation impairment, or because of malicious attacks on the network.
- (4) **Availability:** The network should remain operational all the time. It must be robust enough to tolerate link failures and also be capable of surviving various attacks mounted on it. It should be able to provide the guaranteed services whenever an authorized user requires them.
- (5) **Non-repudiation:** Non-repudiation is a mechanism to guarantee that the sender of a message cannot later deny having sent the message and that the recipient cannot deny having received the message. Digital signatures (DS), which function as unique identifiers for each user, much like a written signature, are used commonly for this purpose.

1.2.2 Challenges and Limitations to MANETs Security

Where in wired networks, an adversary must gain physical access to the wired link or sneak through security holes at firewalls and routers, wireless attacks may come from anywhere along all directions. MANETs have no clear line of defense and every node must be prepared for encounters with an adversary. Therefore, some special security measures must be considered to enhance the security level of MANETs [Cap 03b].

Security has become a primary concern in MANETs in order to provide protected communication between mobile nodes in a hostile environment.

The unique characteristics of MANETs pose a number of nontrivial challenges to security design. These challenges make a case for building security solutions that achieve both broad protection and desirable network performance.

Security design in such infrastructureless wireless MANETs is challenging for several reasons [Kon 01]:

- Security breach: Wireless transmissions are prone to security attacks, and it is very likely that adversaries will eventually break into a limited number of entities over a large time window.
- Mobility and service ubiquity: Mobile users incur dynamic topological changes. A mobile user may be able to perform effective and timely communication with its local neighbors but not with remote entities.
- Network dynamics: Channel errors and node failures all incur dynamics into the network. Besides, an entity may join and leave the network over time.
- Network scale: The number of networking devices can be large, thus a scalable solution is critical.

MANETs are very vulnerable to a number of security attacks, such as [Luo 02]:

- Passive eavesdropping over wireless channel.
- Denial-of-Service (DoS) attacks by malicious nodes.
- Attacks from compromised entities or stolen devices.

1.3. Authentication

1.3.1 Definitions

Authentication is the verification of the identity of a party who generated some messages, and of the integrity of the messages. In computer networks, two types of authentication can be identified, namely, message authentication and node

authentication. Message authentication is a technique for verifying the integrity of a transmitted message. While node authentication is a technique to let one party prove the identity of another party. There are two differences between message authentication and node authentication; these are [For 08]:

- (1) Message authentication may not happen in real time; node authentication does. In message authentication, when a sender sends a message to a receiver, while the receiver authenticates the message; the sender may or may not be present in the communication process. On the other hand, when the sender requests node authentication, there is no real message communication involved until the sender is authenticated by the receiver. The sender needs to be online and takes part in the authentication process. Only after the sender is authenticated can the message be communicated between the two parties.
- (2) Message authentication simply authenticates one message; the process needs to be repeated for each new message. Node authentication authenticates the sender for the entire duration of a session.

In this thesis, we are concerned with node authentication. In node authentication the sender must identify itself to the receiver. This can be done with one of the three kinds of witnesses; something known, something possessed, or something inherent [Bra 06].

- (1) Something known. This is a secret known only by the sender that can be checked by the receiver. Examples are password, a PIN number, a secret key, and private key.
- (2) Something possessed. This is something that can prove the sender identity. Examples are a passport, a driver's license, an identification card, a credit card, and a smart card.

- (3) Something inherent. This is an inherent characteristic of the sender. Examples are conventional signature, fingerprints, voice, facial characteristics, retinal pattern, and handwriting.

1.3.2 Authentication Techniques

A number of node authentication techniques have been developed throughout the years, such as: password-based authentication, challenge-response authentication, Kerberos-based authentication, public-key cryptography, etc. In what follows a brief description is provided for each of them.

Password-based authentication

The simplest and the oldest method of node authentication is the password, something that the sender possesses. A password is used when a user needs to access a system to use its resources (login). Each user has a user identification that is public and a password that is private. This authentication scheme is divided into two separate groups: the fixed password and the one-time password.

Challenge-response authentication

In this type of authentication the sender proves that it knows a secret without actually sending it. In this scheme, the challenge is a time varying value sent by the receiver, and the response is the result of a function applied to the challenge. It can be divided into four categories: symmetric-key ciphers, keyed-hash functions, asymmetric-key ciphers, and digital signature.

Kerberos-based authentication

Kerberos [Neu 05, Koh 94, Koh 93] is a distributed authentication service that allows a sender (client) to prove its identity to a receiver (an application server, or just server) without sending data across the network that might allow an attacker or the receiver to subsequently impersonate the principal. Kerberos optionally

provides integrity and confidentiality for data sent between the sender and receiver. Kerberos was developed in the mid-'80s as part of MIT's Project Athena [Cha 90].

Kerberos is not effective against password guessing attacks; if a user chooses a poor password, then an attacker guessing that password can impersonate the user. Similarly, Kerberos requires a trusted path through which passwords are entered. If the user enters a password to a program that has already been modified by an attacker (a Trojan horse), or if the path between the user and the initial authentication program can be monitored, then an attacker may obtain sufficient information to impersonate the user.

To be useful, Kerberos must be integrated with other parts of the system. It does not protect all messages sent between two nodes; it only protects the messages from software that has been written or modified to use. While it may be used to exchange encryption keys when establishing link encryption and network level security services, this would require changes to the network software of the hosts involved.

Kerberos does not itself provide authorization, but V5 Kerberos passes authorization information generated by other services. In this manner, Kerberos can be used as a base for building separate distributed authorization services [Neu 05].

The Kerberos authentication system uses a series of encrypted messages to prove to a receiver that a client is running on behalf of a particular sender. The Kerberos protocol is based in part on the Needham and Schroeder authentication protocol, but with changes to support the needs of the environment for which it was developed. Among these changes are the use of timestamps to reduce the number of messages needed for basic authentication, the addition of a "ticket-granting" service to support subsequent authentication without re-entry of a principal's password, and different approach to cross-realm authentication

(authentication of a sender registered with a different authentication server than the receiver). Further information on Kerberos can be found in RFC 1510 for a more thorough description of the Kerberos protocol [Koh 94].

Public-key cryptography

In public-key cryptography, encryption and decryption are performed using a pair of keys such that knowledge of one key does not provide knowledge of the other key in the pair [For 08, For 07, Sta 03]. One key is published and is called the public key, and the other key is kept private. Public-key cryptography has several advantages over conventional cryptography when used for authentication. These include more natural support for authentication to multiple recipients, support for non-repudiation (since the receiver does not know the private key, it can not generate a message that purports to be from the authenticated sender), and the elimination of secret encryption keys from the central authentication server.

Public-key cryptography is well suited for use in authentication in store and forward applications such as electronic mail, and it is required by applications where a signature is verified by many readers. The most accepted algorithm for public-key cryptography is the RSA algorithm, which was proposed by R. Rivest, A. Shamir, and L. Adleman [Riv 78]. However, message encryption or decryption using the RSA algorithm is time consuming or expensive process.

Public-key encryption may also be used by authentication servers to exchange conventional cross-realm keys on-demand between authentication servers, with the cost amortized over many requests.

Threshold secret sharing

Popular network authentication architectures, such as Kerberos [Neu 05], the X.509 standard [Are 00], and PKI trust model [Per 99], are based on using a globally trusted certification authority (CA) model. However, using a globally trusted CA model may work well in wired or infrastructure (AP) wireless networks

. But, it does not work well in MANETs environments for several reasons:

- (1) MANETs provide no infrastructure support. The cost of maintaining such CA may be prohibitively high.
- (2) Each of the CA servers is exposed to a single point of compromises and failures.
- (3) Multihop communications over the error-prone wireless channel expose data transmissions to high loss rate and large latency.
- (4) Frequent route changes induced by mobility also make locating and contacting CA servers in a timely fashion non-trivial.

Variations of the CA model, such as hierarchical CAs and CA delegations can ameliorate, but cannot address issues such as service availability and robustness [Kon 01, Per 99]. Therefore, more efficient and reliable solutions are required to address the above issues. One alternative solution is to use the concept of threshold secret sharing (TSS) scheme proposed by Adi Shamir in 1978 [Riv 78, Bur 94].

Shamir's TSS scheme is working as follows: a secret (S) can be divided into a number of pieces (say n , where n is the number of nodes within the network), in such a way that S can be reconstructed from any number of pieces (say k), but even complete knowledge $k-1$ pieces is not enough to reconstruct S . This approach can be used to design self-securing networks, in which multiple nodes (k) collaboratively serve as a CA server. Therefore, the authority and functionality of the authentication server are distributed to each node's locality. Any local k nodes are trusted as a whole and collaboratively provide authentication services.

Some pleasant features of this scheme are as follows. The system is not exposed to any single point of compromise, single point of DoS attack, or single point of

failure. Authentication can be performed in every network neighborhood; this feature is important to authenticate roaming users in a MANET. Furthermore, this solution scales to large network size.

1.4. Wireless Network Environments

The wireless network environment can be categorized, according to the presence of noise or packet-loss, into two types of environments; these are [Jar 07]:

- Noiseless (error-free) environment
- Noisy (error-prone) environment

1.4.1 Noiseless (Error-Free) Environment

Noiseless (error-free) environment represents an ideal network environment, in which it is assumed that all data transmitted by a source node is successfully and correctly received by a destination node. It can be characterized by the following axioms or assumptions:

- (1) The world is flat
- (2) All radios have equal range, and their transmission range is circular
- (3) Communication link symmetry
- (4) Perfect link
- (5) Signal strength is a simple function of distance.

1.4.2 Noisy (Error-Prone) Environment

Noisy (error-prone) environment represents a realistic network environment, in which the received signal will differ from the transmitted signal, due to various transmission impairments, such as:

- (1) Wireless signal attenuation (p_{att})
- (2) Free space loss (p_{free})
- (3) Thermal noise (p_{therm})
- (4) Atmospheric absorption (p_{atm})

- (5) Multipath effect (p_{mult})
- (6) Refraction (p_{ref})

All of these impairments are represented by a generic name, noise. The environment is called noisy environment. For modeling and simulation purposes, the noisy environment can be described by introducing a probability function, which is referred to as the probability of reception (p_c). It is defined as the probability that a wireless transmitted data is being lost and successfully delivered to a destination node despite the presence of all or any of the above impairments. Thus, p_c can be calculated as:

$$p_c = p_{att} \cdot p_{free} \cdot p_{therm} \cdot p_{atm} \cdot p_{mult} \cdot p_{ref} \dots\dots \quad (1.1)$$

1.5.Problem Statement

A TSS-based authentication scheme is the most suitable and widely-used scheme for providing self-securing wireless ad hoc networks. A number of researches have been carried-out to develop and investigate its performance in terms of authentication success ratio, average delay, overheads, average number of failures, etc. However, all of these investigations have considered noiseless (error-free) wireless ad hoc network environments. However, in practice, wireless ad hoc networks suffer from high packet-loss due to the presence of noise and node mobility, which may significantly affect the performance of this scheme. Furthermore, the literature is short of clear quantitative investigations on the variation of the performance of the TSS scheme with a number of network parameters, such nodes densities, nodes speeds, nodes radio transmission ranges.

This research is carried-out to develop, implement, and evaluate the performance of the TSS-based authentication scheme in noisy MANETs environments and also investigate the effect of the above mentioned network parameters on the performance of the scheme.

1.6. Research Objectives

The main objectives of this work can be summarized as follows:

- (1) Develop a distributed TSS scheme, which will be referred to as TSS scheme, by utilizing Shamir's TSS concept to control mobile nodes authentication in noisy MANETs. The scheme is implemented on MANSim network simulator.
- (2) Evaluate the performance of the TSS authentication scheme. The performance is evaluated by estimating the variation of the authentication success ratio with the threshold secret share for various nodes densities, nodes speeds, nodes radio transmission ranges, and network noise-levels.
- (3) Examine the effect of each of the above parameters on the performance of the TSS scheme and identify the most effective parameters to be carefully monitored during network operation to facilitate network management.

1.7. Thesis Organization

This chapter provides an introduction to the general domain of this thesis pertaining to: MANET definition, applications, challenges, and limitations. This chapter discusses the main requirements, challenges, and limitations for building secure MANETs. Authentication and authentication techniques that are currently in use and their limitations are also discussed. The rest of this thesis is organized as follows.

Chapter 2 reviews some of the most recent work that is related to node authentication in infrastructure and infrastructureless wireless networks. Chapter 3 presents a description of the TSS scheme. It discusses issues of the formalization of the localized trust model that lays the foundation for the design, expands the adversary model that the system should handle, proposes refined

localized certification services, and develop a new scalable solution of share updates to resist more powerful adversaries. Also, this chapter briefly describes the MANSim network simulator as it is used as platform to implement the TSS scheme.

Chapter 4 presents a description of four scenarios that are performed to evaluate the performance of the TSS scheme in noisy MANETs. These scenarios investigate the effect of a number of networks parameters (e.g., nodes densities, nodes velocities, nodes radio transmission ranges, noise-levels expressed in probability of packet reception). The results for these four scenarios are presented in tables and graphs. Also, in this chapter, the results obtained are discussed.

Finally, in Chapter 5, based on the results obtained from the different scenarios, conclusions are drawn, and suggestions and recommendations for future work are pointed-out.

CHAPTER 2 LITERATURE REVIEWS

A mobile ad hoc network (MANET) is a self-configurable network with dynamic topologies. All nodes in the network share the responsibility for routing, access, and communications [Ban 06]. The MANET can be considered as a short-lived collection of mobile nodes communicating with each other. Such networks are more vulnerable to security threats than traditional wireless networks because of the absence of the fixed infrastructure [Luo 02]. In order to provide secure communications in such networks, lots of mechanisms have been proposed since the early 1990s, which also have to deal with the limitations of the MANETs, including power and bandwidth resources [Lee 07].

One of the major security issues in MANETs is node authentication. Popular network authentication architectures include Kerberos [Neu 05, Koh 94, Koh 93], X.509 standard [Are 00], and public key infrastructure (PKI) [Per 99], are based on using a globally trusted certification authority (CA) model. However, using a globally trusted CA model may work well in wired or infrastructure wireless networks. But, it does not work well in MANETs.

MANETs should provide authentication and key management without a trusted third party (TTP) because of their self-organizing and self-configuring characteristics. Several solutions to this problem have been proposed in MANETs [Hua 08, Akb 08, Bal 02, Luo 02, Kon 01].

In what follows a review of some of the most recent work that is related to node authentication in MANETs, are presented in chronological order. For each year, the reviewed work is ordered according to alphabet of the name of the first author. We would like to emphasize that throughout the literatures that have been reviewed; we have not found any work which investigates the effect of nodes velocities, nodes radio transmission range, and noise-level on the performance of TSS-based authentication schemes in MANETs.

R. Akbani et. al. [Akb 08] studied packet authentication in wireless networks and proposed a hop-by-hop efficient authentication protocol (HEAP). HEAP authenticates packets at every hop by using a modified hash message authentication code (HMAC) based algorithm along with two keys and drops any packets that originate from outsiders. HEAP can be used with multicast, unicast or broadcast applications. Several simulations were performed to compare HEAP with existing authentication schemes, such as timed efficient stream loss-tolerant authentication (TESLA) [Per 02], lightweight hop-by-hop authentication protocol (LHAP) and Lu and Pooch's algorithm [Zhu 06, Zhu 03]. They also measured metrics such as latency, throughput, packet delivery ratio, CPU and memory utilization and showed that HEAP performs very well compared to other schemes while guarding against outsider attacks.

D. Huang and D. Medhi [Hua 08] presented a secure group key management scheme for hierarchical MANETs. The scheme aimed to improve both scalability and survivability of group key management for large-scale MANETs. They proposed two schemes: (1) a multi-level security model, which follows a modified Bell-La Padula security model that is suitable in a hierarchical MANETs, and (2) a decentralized group key management infrastructure to achieve such a multi-level security model. The approaches reduce the key management overhead and improve resilience to any single point failure problem. In addition, they developed a roaming protocol that is able to provide secure group communication involving group members from different groups without requiring new keys; an advantage of this roaming protocol is that it is able to provide continuous group communication even when the group manager fails.

S. Hussain and H. Al-Bahadili [Hus 08] presented a non-exchanged password scheme for password-based authentication in client-server systems. This scheme constructs a digital signature (DS) that is derived from the user password. The DS is then exchanged instead of the password itself for the purpose of authentication. Therefore, the scheme was referred to it as a password-based digital signature

(PBDS) scheme. It consists of three phases, in the first phase a password-based permutation (P) is computed using the key-based random permutation (KBRP) method [Hus 06]. The second phase utilizes P to derive a key (K) using the password-based key derivation (PBKD) algorithm [Hus 06]. The third phase uses P and K to generate a DS to be exchanged between the two communicating parties. The scheme has a number of features that show its advantages over other password authentication approaches.

J. Kim and S. Bahk [Kim 08] designed architecture of mesh certification authority (MeCA) for wireless mesh networks (WMNs). In MeCA, the secret key and functions of CA are distributed over several mobile routers. For secret sharing and redistribution, they develop a fast verifiable share redistribution (FVSR) scheme, which works for threshold cryptography and minimizes the possibility of secret disclosure when some shareholders are compromised by adversaries. MeCA adopts the multicasting based on Ruiz tree, which is optimal in reducing the operation overhead. It can update, revoke, and verify certificates of WMN nodes in a secure and efficient manner. Simulation results showed that MeCA does not disclose its secret key even under severe attacks while incurring low overhead compared to other existing schemes in MANETs.

C. Li et. al. [Li 08] proposed a lightweight authenticated key establishment scheme with privacy preservation to secure the communications between mobile vehicles and roadside infrastructure in a vehicles ad hoc network (VANET), which is called SECSPP. The proposed scheme not only accomplishes vehicle-to-vehicle and vehicle-to-roadside infrastructure authentication and key establishment for communication between members, but also integrates blind signature techniques into the scheme in allowing mobile vehicles to anonymously interact with the services of roadside infrastructure. They also showed that the SECSPP scheme is efficient in its implementation on mobile vehicles in comparison with other related proposals.

A. Mukherjee et. al. [Muk 08] proposed an entirely decentralized key generation mechanism, employing a central trusted entity only during initialization. They showed that using their approach, keys can be established between group members with absolutely no prior communication. The approach relies on threshold cryptography and introduces a novel concept of node-group-key (NGK) mapping. They provided an extensive analytical model for the computations involved and communication costs and also provided a “lie” detection mechanism. The simulation results showed appreciable performance improvement and enhanced robustness.

N. W. Wang et. al. [Wan 08] reviewed the secure infrastructure of VANET, some potential applications and interesting security challenges. To cope with these security challenges, they proposed a secure scheme for vehicular communication on VANETs. The proposed scheme not only protects the privacy but also maintains the liability in the secure communications by using session keys. They also analyzed the robustness of the proposed scheme.

C. Y. Yeun et. al. [Yeu 08] proposed a novel authenticated group key agreement protocol for end-to-end security in the MANET environment without any infrastructure that is based on Burmester and Desmedt (BD) group key agreement protocol [Bur 05, Bur 94] and its variants [Cho 04]. They also designed practical enhancements of these protocols that not only detect, but also identify malicious insiders by using the trusted arbiter who is only involve in the protocol if cheating has occurred.

Z. Chai et. al. [Cha 07] proposed a threshold password authentication scheme, which meets both availability and strong security requirements in MANETs. In this scheme, t out of n server nodes can jointly achieve mutual authentication with a registered user within only two rounds of message exchanges. The scheme allows users to choose and change their memorable password without subjecting to guessing attacks. Moreover, there is no password table in the server nodes end,

which is preferable since mobile nodes are usually memory-restricted devices. They also showed that the scheme is efficient to be implemented in mobile devices.

N. Komninos et. al. [Kom 07, Kom 06] demonstrated that when security of a given network architecture is not properly designed from the beginning; it is difficult to preserve confidentiality, authenticity, integrity and non-repudiation in practical networks. They also investigated the principal security issues for protecting MANETs at the data link and network layers. The security requirements for these two layers are identified and the design criteria for creating secure ad hoc networks using multiple lines of defense against malicious attacks are discussed. The performance of several challenge–response based protocols, was presented and analyzed through simulation results.

J. S. Lee and C. C. Chang [Lee 07] proposed an ID-based version of the secure communication scheme for cluster-based ad hoc networks based on PKI, which had been developed by Varadharajan et. al. [Var 04], for providing secure communications in ad hoc networks. Varadharajan et. al. scheme suffers from huge computation overheads invoked by the PKI cryptosystem for each communicating node in the cluster. Lee et. al. scheme eliminates the need for adopting PKI cryptosystems; therefore, computation overheads of involved nodes in the their scheme can be reduced by 25% at least.

Z. Li and J. J. Garcia-Luna-Aceves [Li 07] presented a non-interactive key agreement and progression (NIKAP) scheme for MANETs, which does not require an on-line centralized authority. It can non-interactively establish and update pairwise keys between nodes, and it is configurable to operate synchronously or asynchronously. It also supports differentiated security services with respect to the given security policies. NIKAP is valuable to scenarios where pairwise keys are desired to be established without explicit negotiation over insecure channels, and also need to be updated frequently.

G. Wang et. al. [Wan 07a] introduced some existing rekeying schemes for secure multi-privileged group communications and analyzed their advantages and disadvantages. Then, they proposed an efficient group key management scheme called ID-based hierarchical key graph scheme (IDHKGS) for secure multi-privileged group communications. The IDHKGS scheme employs a key graph, on which each node is assigned a unique ID according to access relations between nodes. When a user joins/leaves the group or changes its access privileges, other users in the group can deduce the new keys using one-way function by themselves according to the ID of joining/leaving/changing node on the graph, and thus the IDHKGS scheme can greatly reduce the rekeying overhead.

N. C. Wang and S. Z. Fang [Wan 07b] proposed a hierarchical key management scheme (HKMS) for secure group communications in MANETs. For the sake of security, in this scheme a packet is encrypted twice. Furthermore, due to the frequent changes of the topology of a MANET, the group maintenance was discussed. A number of simulation were carried-out to compare the performance of the HKMS scheme with two different protocols, these are:

- The secure group communication protocol for ad hoc wireless networks proposed by Y. M. Tseng et. al. [Tse 07].
- The group key agreement approach proposed by Steiner et. al. [Ste 98].

B. Wu et. al. [Wu 07] proposed a secure and efficient key management (SEKM) framework for MANETs. SEKM builds a PKI by applying a secret sharing scheme and using an underlying multi-cast server groups. They gave detailed information on the formation and maintenance of the server groups. In SEKM, each server group creates a view of the CA and provides certificate update service for all nodes, including the servers themselves. A ticket scheme was introduced for efficient certificate service. In addition, an efficient server group updating scheme was proposed. The performance of SEKM was evaluated through simulation.

M. Abdalla et. al. [Abd 06] designed a password-based authenticated key exchange protocol to secure networks even when the secret key or password shared between two users is drawn from a small set of values. Due to the low entropy of passwords, such protocols are always subject to online guessing attacks. In these attacks, the adversary may succeed with non-negligible probability by guessing the password shared between two users during its on-line attempt to impersonate one of these users. The main goal of password-based authenticated key exchange protocols is to restrict the adversary to this case only.

In their protocol, they consider password-based authenticated key exchange in the three-party scenario, in which the users trying to establish a secret do not share a password between themselves but only with a trusted server. In order to achieve their objective of developing a secure protocol, they recalled some of the existing security notions for password-based authenticated key exchange protocols and introduce new ones that were more suitable to the case of generic constructions. They then presented a natural generic construction of a three-party protocol, based on any two-party authenticated key exchange protocol, and proved its security without making use of the random Oracle model. Their modified protocol was the first provably-secure password-based protocol in the three-party setting.

S. Zhu et. al. [Zhu 06, Zhu 03] proposed a lightweight hop-by-hop authentication protocol (LHAP) for ad hoc networks. LHAP resides in between the network layer and the data link layer, thus providing a layer of protection that can prevent or thwart many attacks from happening, including outsider attacks and insider impersonation attacks. LHAP is based on two techniques: (1) hop-by-hop authentication for verifying the authenticity of all the packets transmitted in the network and (2) one-way key chain and TESLA for packet authentication and for reducing the overhead for establishing trust among nodes. They analyzed the security of LHAP, and show LHAP is a lightweight security protocol through detailed performance analysis. Their detailed performance evaluation showed that LHAP incurs small performance overhead and it also allows a tradeoff between security and performance.

B. Lu and Udo W. Pooch [Lu 05] proposed a lightweight authentication protocol, which utilizes one-way hash chain to provide effective and efficient authentication for communications between neighboring nodes in MANETs. Delayed key disclosure scheme was used to prevent in-the-middle attack on key release. The security properties of the protocol were analyzed. They also demonstrated simulation results and performance analysis on trust management, message authentication and the delayed key disclosure approach. The analysis showed that the protocol incurs low overhead penalty and achieves a low dropped packet rate on key disclosure with a cache of fair size.

B. Zhu et. al. [Zhu 05] proposed a novel hierarchical scheme based on threshold cryptography to address both security and efficiency issues of key management and certification service in MANET. The main contributions of Zhu et. al. key management scheme include: (1) providing various parts of MANET the flexibility of selecting appropriate security configurations, according to the risks faced; (2) providing the adaptivity to cope with rapidly-changing environments; (3) handling of MANETs with a large number of nodes; (4) issuing certificates with different levels of assurance. They also proposed two algorithms, which can be used independently from the hierarchical structure to protect certification services in ad hoc networks from active attacks. Their simulation results show that, compared to the previous work, their second algorithm is much faster in a friendly environment. Simulation results also showed that the two algorithms work well in a hostile environment in which existing schemes work poorly.

S. Zhu et. al. [Zhu 04] presented an efficient and scalable group rekeying protocol for secure multicast in ad hoc networks, namely, GKMPAN. The protocol exploits the property of ad hoc networks that each member of a group is both a host and a router, and distributes the group key to member nodes via a secure hop-by-hop propagation scheme. A probabilistic scheme based on pre-deployed symmetric keys is used for implementing secure channels between members for group key distribution. GKMPAN also includes a novel distributed scheme for efficiently

updating the pre-deployed keys.

GKMPAN has three attractive properties. First, it is significantly more efficient than group rekeying schemes that were adapted from those proposed for wired networks. Second, GKMPAN has the property of partial statelessness; that is, a node can decode the current group key even if it has missed a certain number of previous group rekeying operations. This makes it very attractive for ad hoc networks where nodes may lose packets due to transmission link errors or temporary network partitions. Third, in GKMPAN the key server does not need any information about the topology of the ad hoc network or the geographic location of the members of the group. They also studied the security and performance of GKMPAN through detailed analysis and simulation; we had also implemented GKMPAN in a sensor network test-bed.

S. Capkun et. al. [Cap 03a, Cap 03b] demonstrated that for many reasons, traditional security solutions that require on-line trusted authorities or certificate repositories are not well suited for securing ad hoc networks. So that they proposed a fully self-organized public-key management system that allows users to generate their public-private key pairs, to issue certificates, and to perform authentication regardless of the network partitions and without any centralized services. Furthermore, the approach they proposed does not require any trusted authority, not even in the system initialization phase. The key idea is that if two nodes are in the vicinity of each other, they can establish a security association (SA) by exchanging appropriate cryptographic material through a secure channel with the short transmission range. However, this direct solution takes a long time because it requires a node to encounter every node that it wants to communicate with.

M. Narasimha et. al. [Nar 03] explored the use of threshold cryptography in peer-to-peer settings (both Internet and MANET) to provide, in a robust and fault tolerant fashion, security services such as authentication, certificate issuance and access

control. Threshold cryptography provides high availability by distributing trust throughout the group and is, therefore, an attractive solution for secure peer-groups. It seems so, at least. Their work investigated the applicability of threshold cryptography for membership control in peer-to-peer systems. In the process, they discovered that one interesting proposed scheme contains an unfortunate (yet serious) flaw. Then, they presented an alternative solution and its performance measurements. More importantly, their work cast a certain degree of skepticism on the practicality and even viability of using (seemingly attractive) threshold cryptography in certain peer-to-peer settings.

D. Balfanz et. al. [Bal 02] addressed the problem of secure communication and authentication in ad-hoc wireless networks, and presented a user-friendly solution, which provides secure authentication using almost any established public-key-based key exchange protocol, as well as inexpensive hash-based alternatives. In their approach, devices exchange a limited amount of public information over a privileged side channel, which will then allow them to complete an authenticated key exchange protocol over the wireless link. Their solution does not require a public key infrastructure, is secure against passive attacks on the privileged side channel and all attacks on the wireless link, and directly captures users' intuitions that they want to talk to a particular previously unknown device in their physical proximity. They had implemented their approach in Java for a variety of different devices, communication media, and key exchange protocols.

H. Luo et. al. [Luo 02] demonstrated that a centralized or hierarchical network security solution does not work well in MANETs, so that they proposed a scalable, distributed authentication services in MENETs. Their design took a self-securing approach, in which multiple nodes (say, k nodes) collaboratively provide authentication services for other nodes in the network. They first formalized a localized trust model that lays the foundation for the design, refined localized certification services, developed a scalable share update to resist more powerful adversaries, and finally, they evaluated their solution through simulation and

implementation. The results they obtained showed that the proposed algorithm ensures an excellent performance and can facilitate practical deployment in a potentially large-scale network with dynamic node membership.

A. Perrig et. al. [Per 02] presented, implemented, and evaluated the performance of an efficient protocol with low communication and computation overhead, which scales to large numbers of receivers, and tolerates packet-loss, namely, TESLA broadcast authentication protocol. TESLA is based on loose time synchronization between the sender and the receivers. Despite using purely symmetric cryptographic functions (Message Authentication Code (MAC) functions), TESLA achieves asymmetric properties. The main idea of TESLA is that the sender attaches to each packet a MAC computed with a key K known only to itself. The receiver buffers the received packet without being able to authenticate it. A short while later the sender discloses K and the receiver is able to authenticate the packet. Consequently, a single MAC per packet suffices to provide broadcast authentication, provided that the receiver has synchronized its clock with the sender ahead of time.

J. P. Hubaux et. al. [Hub 01] proposed a scheme based on a chain of public-key certificates, which is scalable and self-organized. However, the scheme does not guarantee authentication service between any two nodes even though they are in the same secure domain, but provides only probabilistic guarantee. There is also a storage problem because each node has to store relatively many other nodes' certificates.

J. Kong et. al. [Kon 01] described a design that supports ubiquitous security services for mobile hosts, scales to network size, and is robust against break-ins. In the design, the certification authority functions are distributed through a threshold secret sharing mechanism, in which each entity holds a secret share and multiple entities in a local neighborhood jointly provide complete services. Furthermore, in this design, localized certification schemes are employed to enable

ubiquitous services. Also the secret shares are continuously updated to further enhance robustness against break-ins. A number of scenarios were simulated to demonstrate and confirm the effectiveness of the design.

R. Canetti et. al. [Can 99] constructed a broadcast authentication protocol based on k different keys to authenticate every message with k different MAC's. Every receiver knows m keys and can hence verify m MAC's. The keys are distributed in such a way that no coalition of w receivers can forge a packet for a specific receiver. The security of their scheme depends on the assumption that at most a bounded number (which is on the order of k) of receivers collude.

L. Zhou and Z. J. Hass [Zho 99] identified the vulnerability of using a centralized CA for authentication in ad-hoc networks and proposed a method with multiple CAs based on threshold cryptography [Des 94]. These multiple CAs have secret shares of a certificate authority signing key (CASK) while no CAs individually know the whole complete CASK, which can be known only when CAs of more than k collaborate. Therefore, this method can support the network security against up to $k-1$ collaborative compromised nodes. While Zhou and Hass's method improved the robustness of the authentication system, it depended on the offline authority which elects q CAs ($q \geq k$) during the bootstrapping phase. Furthermore, it has poor availability because if $q-k+1$ CAs have been compromised, uncompromised $k-1$ CAs that are left can not provide authentication services anymore.

CHAPTER 3

THE THRESHOLD SECRET SHARING (TSS) SCHEME

This chapter presents the concept and the overall architecture and implementation of the proposed threshold secret sharing (TSS) scheme for self-securing noisy mobile ad hoc networks (MANETs). The proposed TSS scheme is based on the well-known concept of threshold secret share proposed by Adi Shamir in 1979 [Sha 79]. This concept is proved to be very efficient and reliable to be used for self-securing MANETs.

Shamir's TSS scheme is working as follows: a secret (S) can be divided into a number of pieces (say n , where n is the number of nodes within the network), in such a way that S can be reconstructed from any number of pieces (say k), but even complete knowledge of $k-1$ pieces is not enough to reconstruct S . This approach can be used to design self-securing networks, in which multiple nodes (k) collaboratively serve as a certification authority (CA) server. Therefore, the authority and functionality of the authentication server are distributed to each node's locality [Bur 94].

This scheme is characterized by many important features, such as: the system does not expose to any single point of compromise, single point of denial-of-service (DoS) attack, or single point of failure. Authentication can be performed in every network neighborhood; this feature is important to authenticate roaming users in MANETs. Furthermore, this solution can scale well to large network size [Kon 01].

Section 3.1 presents a description of the RSA asymmetric public-key encryption algorithm, which is used to ensure secure exchange of the shares between collaborative nodes. The authentication models that have been developed throughout the years are discussed in Section 3.2 with more emphasis on two widely-used models, namely, the trusted third party (TTP) model and the localized-trust model. The concept of Shamir threshold secret share is discussed in Section

3.3. In Section 3.4, we provide a detailed description of the proposed TSS scheme and its localized-trust certification procedure, which is based on the standard RSA asymmetric public-key security algorithm.

The system, adversary, and intrusion models are described in Section 3.5. Section 3.6, gives a narrative for the network simulator (MANSim) that is used as simulation platform in this thesis. In particular, we present a description for one of its modules, the computation module. The parameters that can be used to evaluate the performance of the TSS scheme are defined in Section 3.7. Finally, in section 3.8, the practical implementation of the TSS scheme is discussed.

3.1 .RSA Asymmetric Public-Key Security Algorithm

Public-key cryptography is asymmetric cryptography proposed by Whitfield Diffie and Martin Hellman at Stanford University in 1976 [Dif 76]. It involves the use of two keys:

- A public-key, which may be known by anybody, and can be used to encrypt messages and/or verified signatures.
- A private-key, which is known only to the recipient, and used to decrypt messages and/or signed (created) signatures.

It is called asymmetric cryptography because the key that is used to encrypt messages or verify signatures cannot decrypt messages or create new signatures. The use of public-key cryptography can be classified into three broad categories; these are [For 08, Sta 03]:

- Encryption/decryption to provide data security.
- Digital signatures to provide message authentication.
- Session keys exchange.

There are a number of public-key algorithms that have been developed. Some are suitable for all uses, others algorithms are only adequate for a particular application. In addition to relying on two keys, public-key algorithms have the following characteristics:

- Computationally infeasible to find the decryption key by knowing only the algorithm and the encryption key.
- Computationally easy to encrypt/decrypt messages when the relevant encrypt/decrypt key is known.

Either of the two related keys can be used for encryption, with the other used for decryption (in some schemes). However, public-key cryptanalysis is theoretically possible using brute force exhaustive search attack. But, as it is well-known that using too large keys (>512 bits) makes brute force attack impractical. On the other hand, using too large keys makes public-key algorithms slow compared to private key schemes.

One of the best known and widely used public-key algorithms is the RSA algorithm, which came after the names of three scientists, namely, Ron Rivest, Adi Shamir, and Len Adleman of MIT [Riv 78].

The RSA algorithm is a block cipher in which plaintext is encrypted in blocks, with each block having a decimal value less than specific prime number (m) which is also referred to as the modulus. That is, the block size b must be less than or equal to $\log_2(m)$ (i.e., $b \leq \log_2(m)$), and a typical block size b is 1024 bits. In practice, a block size of s bits, where $2^s < m \leq 2^{s+1}$ is used. Encryption and decryption can be expressed mathematically as follows:

$$C = M^E \text{ mod } m \quad (3.1)$$

$$M = C^D \text{ mod } m = (M^E)^D \text{ mod } m = M^{ED} \text{ mod } m \quad (3.2)$$

Where

- M is the plaintext block (message) of b -bit length and it has an integer value between 2^0 and 2^b-1 ,
- C is the ciphertext block of b -bit length and it has an integer value between 2^0 and 2^b-1 ,
- E is the encryption key (public-key),
- D is the decryption key (private-key),
- m is the modulus,
- b is the block size in bits.

The public-key encryption algorithm usually has a public key of $KU=\{E, m\}$ and a private key $KR=\{D, m\}$. For this algorithm to be satisfactory for public-key cryptography, the following requirements must be kept:

- It is possible to find values of E, D, m such that $M^{ED}=M \pmod m$ for all $M < m$.
- It is relatively easy to calculate M^E and C^D for all values of $M < m$.
- It is infeasible to determine D given E and m .

Each user generates a public/private key pair by:

- Selecting two large prime numbers at random: p and q .
- Computing their system modulus $m=p \cdot q$.
- Compute $Z=(p-1)(q-1)$ which equivalent to the Euler totient function ($\phi(m)$).
- Selecting at random the encryption key E , where $1 < E < Z$, $\text{GCD}(E, Z)=1$, where GCD is the greatest common divisor.
- Solve the equation $E \cdot D=1 \pmod Z$ ($0 \leq d \leq m$) to find decryption key D .

- Publish their public encryption key: $KU = \{E, m\}$.
- Keep secret private decryption key: $KR = \{d, m\}$.

3.2 .Authentication Models

A well-defined authentication (trust) model is fundamental in authentication protocols, therefore, a number of models have been developed, such as:

- (1) The trusted third party (TTP) model [Per 99].
- (2) The pretty good privacy (PGP) model [Gar 95].
- (3) The distributed trust model [Abd 97].
- (4) The localized-trust model [Luo 02]

In the TTP model, an entity is trusted (authenticated) only if it is verified by a CA. While implementations of the TTP model possess efficiency and manageability properties in centralized systems, they suffer from scalability and robustness problems. In the PGP model, each entity manages its own trust based on direct recommendation. It was developed by Philip R. Zimmermann in 1991 and can be used to encrypt and decrypt e-mail over the Internet. It can also be used to send an encrypted digital signature that lets the receiver verify the sender's identity and know that the message was not changed en route. A number of distributed trust techniques were proposed to further quantify the notions of trust and recommendation. However, all of these three models were not enough to address the unique security issues in MANETs. Therefore, the localized-trust model was developed to address the unique security issues of MANETs.

In this section, we shall provide a detailed discussion for the first and the forth models, namely, the TTP and the localized-trust models, while discussions of the PGP and the distributed trust models can be found in [Gar 95] and [Abd 97], respectively.

3.2.1 The TTP model

In the TTP model, an entity is authenticated only if it is verified by the system CA. The model is widely used in access point wireless networks, because of its simplicity, efficiency, and manageability. It only provides limited scalability and robustness. Furthermore, the availability of the system CA is one of critical issues, which needs to be carefully considered. There are four different cases for the availability of the system CA:

(1) CA always available

The case that a CA is always accessible by all network nodes is generally not considered as an option in MANETs, because MANETs should be self-organized after their initialization. If a CA is permanently available we could implement solutions that require certificates or implement Kerberos-like solutions where the TTP distributes session keys. However, in the future it might be reasonable to assume internet connection availability in MANETs. In this case we only need to cope with the resource constraints and mobility of the devices.

(2) CA available at network initialization phase and every time a node joins

The second case comprises all scenarios where a CA is available to issue certificates, and generate and distribute key material and system parameters at the initial stage of the network. The CA is also available for all nodes that subsequently join the network in order to obtain the required system parameters and keys. The assumption that a CA is available every time a new node joins the network is not as restrictive as it might sound. The CA does not need to be accessible by all network nodes every time a new node joins a network. There could be implementations in which nodes contact a CA in order to receive the required data, such as a certificate of the public key or a symmetric key, before joining the network.

(3) CA available at network initialization phase

This case is similar to the previous one, with the difference that subsequently added nodes cannot access the CA. After the initialization phase, the CA cannot be contacted anymore by any of the nodes, including the nodes in the networks and newly joining nodes. Usually this is called the self-organization property of the network. The present network nodes are responsible for taking over the tasks of the CA, such as issuing, renewing, and revoking certificates.

(4) No CA available at any network phase

If no CA is available at all and we still want to use public key encryption schemes, the nodes need to issue their own certificates or we need to implement a model that does not require any public key certificates. The first case can be realized by protocols in the self-organization model and the latter case by protocols in the certificateless public key model.

3.2.2 The Localized-trust model

In this model, an entity is trusted if any k trusted entities claim so within a certain time period T . These k entities are typically among the entities of the first-hop neighbors. Once a node is trusted by its local community, it is globally accepted as a trusted node. Otherwise, a locally distrusted entity is regarded as untrustworthy in the entire network. k and T are two important parameters with T characterizing the time-varying feature of a trust relationship.

Two options for setting k are as follows:

- (1) The first is to set k as a globally fixed parameter that is honored by each entity in the system. In this case, k acts as a system-wide trust threshold.
- (2) The second option is to set k as a location-dependent variable. For instance, k may be the majority of each node's neighboring nodes.

It is clear that the second option provides more flexibility to work in concert with diverse local network topology. However, there is no clear system-wide trust criterion. Due to lack of effective mechanisms to authoritatively determine a node's neighborhood in a mobile environment, the adversaries may take the advantage of this feature.

Trust management and maintenance are distributed in both k and T domains in this localized-trust model. This property is particularly appropriate for a large dynamic MANET, where centralized trust management would be difficult or expensive. Besides, a node indeed cares most about the trustworthiness of its immediate neighbors in practice. This is because a node will communicate with the rest of the world via its one-hop neighbors.

3.3. Concept of Shamir's Secret Sharing Scheme

The concept of Shamir's secret sharing scheme is to divide a data D into n pieces (shares) in such a way that D is easily reconstructable from any k shares, but even complete knowledge of $k-1$ shares reveals absolutely no information about D [Sha 79]. The scheme is referred to as (k, n) threshold scheme. This scheme can be used to construct robust key management techniques for cryptographic systems that can function securely and reliably even when security breaches expose to $k-1$ shares. So that instead of having a central or distributed CA to control key distribution, the key can be divided into shares that are distributed between nodes, then each node can locally construct the key after collecting the shares of k nodes.

According to Shamir, a data D can be divided into n shares D_x ($x=1, 2, \dots, n$) using the following polynomial:

$$q(x) = (a_0 + a_1 x + \dots + a_{k-1} x^{k-1}) = a_0 + \sum_{i=1}^{k-1} a_i x^i \quad (3.3)$$

In which $D_x=q(x)$ and $D=a_0$, so that it can be expressed as:

$$\text{Or } D_x = D + \sum_{i=1}^{k-1} a_i x^i \quad (3.4)$$

Then, given any subset of k of these D_x values (together with their identifying indices), the coefficients (a_0 to a_{k-1}) of $q(x)$ can be evaluated by interpolation, and then evaluate $D=q(0)$ or $D=a_0$.

To make the above claim more precise, modular arithmetic is used instead of real arithmetic. Where, in order to make the interpolation possible all integer coefficients are taken as modulo of a prime number p , which is bigger than both n and D . In other words, the integer coefficients a_1 to a_{k-1} are either chosen between 0 and less than p ($0 \leq a_i < p$) or calculated as $a_i = a_i \text{ mode } p$, where $i=1, 2, \dots, k$. Furthermore, the values of D_x ($x=1, 2, \dots, n$) are also computed modulo p . Thus, Eqn. (3.3) are expressed as:

$$D_x = D + \sum_{i=1}^{k-1} a_i x^i \text{ mod } p \quad (3.5)$$

Some of the useful properties of this (k, n) threshold scheme (when compared to the mechanical locks and keys solutions) are:

- (1) The size of each share does not exceed the size of the original data.
- (2) When k is kept fixed, D_x shares can be dynamically added or deleted (e.g., when executives join or leave the network) without affecting the other D_x shares. A piece is deleted only when a leaving executive makes it completely inaccessible, even to himself.
- (3) It is easy to change the D_x shares without changing the original data D - all we need is a new polynomial $q(x)$ with the same free term. A frequent change of this type can greatly enhance security since the pieces exposed by security breaches cannot be accumulated unless all of them are values

- (4) of the same edition of the $q(x)$ polynomial.
- (5) By using tuples of polynomial values as D_x shares, we can get a hierarchical scheme in which the number of shares needed to determine D depends on their importance. For example, if we give each high importance node three values of $q(x)$, each moderate importance node two values of $q(x)$, and each low importance nodes one value of $q(x)$.

3.4. The Proposed TSS Scheme

3.4.1 Concept of the proposed TSS scheme

In distributed architecture, each node carries a certificate signed with SKR . SKU is assumed to be well-known for certificate verification. Nodes without valid certificates are denied from access to any network resources such as routing and packet forwarding. When a mobile node moves to a new location, it exchanges certificates with its new neighbors. Authenticated neighboring nodes help each other forward and route packets. They also monitor each other to detect possible break-ins. Specific monitoring mechanisms are left to each individual node's choice.

Certificates are stamped with expiration time. Nodes have to be issued a new certificate upon the expiration of its old certificate. In the centralized authentication architecture, nodes have to contact a CA server for this service. In MANET architecture, the certificate-signing key SKR is distributed into each node of the network. Node n_i requests new certificate from any coalition of k nodes, typically among its one-hop neighbors. Upon the receipt of n_i 's certification request, a node checks its records. If its record shows n_i as a well-behaving legitimate node, it returns a partial certificate by applying its share of SKR . Otherwise the request is dropped. By collecting k partial certificates, n_i combines them together to generate the full new certificate as if it were from a CA server.

A misbehaving or broken node that is detected by its neighbors will be unable to renew its certificate. It will be cut off from the network at the expiration of its current certificate. A valid certificate in this system represents the trust from a coalition of k nodes. Nodes with valid certificates are globally trusted. Each node contributes to the overall trust management and maintenance by monitoring and certifying its neighboring nodes. The protocol described above has minimum requirements on the reliability of the underlying wireless channel. As long as k neighbors respond, other neighbors are free to move or fail; additional responses may be discarded.

3.4.2 Implementation

As we discussed in Section 3.1, in an RSA-based design, the system CA key pair is denoted as $\{SKR, SKU\}$, where SKR is the system private key and SKU is the system public key. SKR is used to sign certificates for all nodes in the network. A certificate signed by SKR can be verified (decrypted) only by the well-known public key SKU .

Using Shamir threshold secret sharing concept discussed in Section 3.3, SKR is shared among network nodes. Each node n_i holds a secret share $SKR(n_i)$, and any k of such secret share holders can collectively function as the role of CA. However, for better system security, the secrecy of SKR is preserved all the time and it is not visible, known or recoverable by any network node.

Besides the system key pair, each node n_i also holds a personal RSA key pair $\{nkr_i, nku_i\}$. To certify its personal keys, each node n_i holds the certificate C_i in the format of $\langle n_i, nku_i, T \rangle$, which reads as: "It is certified that the personal public key of n_i is nku_i during the time interval $[t, t+T]$ ". A certificate is valid only if it is signed by system secret key SKR .

The proposed TSS scheme makes an extensive use of the polynomial secret sharing scheme due to Shamir, which was discussed in Section 3.3. A secret, specifically the certificate-signing key SKR , is shared among all n nodes in the network according to Eqn. (3.3) with SKR replacing D .

$$SKR(n_i) = (SKR + \sum_{j=1}^{k-1} a_j n_i^j) \bmod p \quad (3.6)$$

Where $SKR(n_i)$ is the node secret share, n_i is the node's ID, SKR is the system private key, k is the minimum number of shares required to recover SKR , n is the total number of nodes within the network, and p is a prime number bigger than n and SKR . In other words, the integer coefficients a_1 to a_{k-1} are either chosen between 0 and less than p ($0 \leq a_j < p$) or calculated as $a_j = a_j \bmod p$, where $i=1, 2, \dots, k$. The same is for SKR , either it is less than p or it is calculated as $SKR = SKR \bmod p$. A coalition of k nodes with k polynomial shares can potentially recover SKR . In fact, there are two cases, these are:

- (1) A newly arrived node or a node that knows its partial share of SKR ($SKR(n_x)$), where n_x is the node ID. In this case, it needs the IDs and shares of $k-1$ nodes to construct k linear equations to solve for SKR .
- (2) A node n_x does not know its partial share $SKR(n_x)$. In this case, the node first needs the IDs and shares of $k-1$ nodes to calculate its share using Lagrange interpolation [Epp 02] as follows:

$$SKR(n_x) = \left(\sum_{j=1}^{k-1} SKR(n_j) \ell_{n_j}(n_x) \right) \bmod p \quad (3.7)$$

where

$$\ell_{n_j}(n_x) = \prod_{\substack{i=1 \\ i \neq j}}^{k-1} \frac{n_x - n_i}{n_j - n_i} \quad (3.8)$$

Then, after having k IDs and shares, a node n_x can construct a set of k linear equations to calculate SKR . In both cases, no coalition up to $k-1$ nodes can yield any information about SKR [Sha 79].

3.4.3 The proposed TSS scheme localized certification procedure

In this scheme, a node n_i firstly locates a coalition B of K neighbors $\{n_1, \dots, n_K\}$ ($K \geq k$) and broadcasts certification requests to them. A node $n_j \in B$ checks its monitoring data on n_i to decide if certification service is granted, then it calculates its partial certificate and sends it back to node n_i . Upon receiving k partial certificates from coalition B , node n_i processes them together to recover its full certificate. Figure (3.1) outlines the main steps of localized certification procedure for the proposed TSS scheme.

// The localized certification procedure of the proposed TSS scheme.
For any node n_i which needs to get a new certificate or to renew its expired certificate: Locates a coalition B of K neighbors $\{n_1, \dots, n_K\}$ ($K \geq k$); Broadcasts certification requests to them; For each node $n_j \in B$ // After receiving certification request Checks its monitoring data on n_i to decide if certification is granted; If (Yes) Then Calculates its partial certificate; Sends it back to node n_i ; Else (No) Discard request; End If Upon receiving k partial certificates from coalition B at node n_i ; Processes them together to recover its new full certificate;

Figure (3.1) - Localized certification procedure of the proposed TSS scheme.

There are two drawbacks in the above approach, these are:

- (1) Firstly, if any node in coalition B fails to respond due to node failures or moving out of range, all the other partial certificates become useless. The computation of other nodes is all wasted and n_i has to restart the whole process from the very beginning.
- (2) The second drawback is that when node n_j receives a certification request from n_i , its records may not provide enough information on n_i . It may be

- (3) because the interaction between n_i and n_j does not last long enough. Moreover, n_i may not exist in n_j 's records at all if they just met. Node n_j has two options in this scenario. One is to serve n_i 's request, since no bad records are located. The risk is that a roaming adversary who cannot get a new certificate from his previous location may take the advantage. The other option is to drop the request, since no records can demonstrate n_i well-behaving. But a legitimate mobile node may not be able to get a new certificate.

However, developing satisfactory solutions to the above two drawbacks is beyond the scope of this thesis, and may be left to future researches. In what follows we discuss how nodes' trustibility is implemented in our simulator.

3.4.4 Trustibility factor (T_f)

The researcher introduces a parameter that simulates the mechanism of misbehaving node identification. This parameter is called network trustibility factor (T_f). It gives a measure of how many nodes within the network trust each other, where each node keeps a T_f value for all other nodes within the network. Its value varies between 0 and 1. If $T_f=1$, a node is considered as a trustable node or a well-behaving node, and all requests will be served immediately. If $T_f \leq 1$, when a recipient node receives a request, it generates a random number (\square), if $\square \leq T_f$, the request will be served, otherwise the node is considered as a misbehaving node and the request is discarded. This approach considerably simplified the identification of a misbehaving node for simulation purposes. Figure (3.2) outlines the procedure for handling misbehaving nodes.

```

// Algorithm for Handling Misbehaving Nodes
// Each node has a database that keeps a record on the nodes trustability factor
( $T_f$ )
    If (A request from a neighboring node is successfully received) Then
        Identify node;
        Search the local database to find if the requesting node has a
        record;
        If (A record is found) Then
            Exclude the  $T_f$  of the requesting node;
        Else
            Set  $T_f$  to 1 ( $T_f=1$ );
        End If
        Generate  $\alpha$ ;          //  $\alpha$  is a random number ( $0 < \alpha \leq 1$ ).
        If ( $\alpha \leq T_f$ ) Then
            Serve request;          // Node is trustable
        Else
            Discard request;        // Node is not trustable
        End If
    End If

```

Figure (3.2) - An algorithm for handling misbehaving nodes.

Practically, this mechanism can be implemented as follows: each node constructs a table or database that keeps a record of an estimated T_f value for each node it communicates with. The value of T_f is estimated according to the history or previous behavior of the communicating node. On the other hand, in order not to let this database grows up in an uncontrolled way, a mechanism for continuously dropping or removing nodes from the database is needed. For example, nodes are dropped or removed from the database if they do not communicate with the targeting node for a pre-defined period of time (interval). The duration of this interval can be adjusted according to the database grown-up rate, size of the available memory, node density, node mobility, etc.

3.5. System, Adversary, and Intrusion Models

3.5.1 System model

In this work, we consider a MANET in which mobile nodes communicate with one another via a bandwidth-constrained, error-prone (noisy), and insecure wireless channel. It is assumed that n mobile nodes are randomly distributed within the

networks area, and n may be dynamically changing as mobile nodes join, leave, or fail over time. The network provides neither physical nor logical infrastructure support, and the reliability of multi-hop packet forwarding based on underlying transport layer and ad hoc routing is not assured. This is implemented in the simulation model by introducing a probability factor, namely, a probability of reception (p_c), which is defined as the probability of an authentication request packet being sent by a sender (source node) will be successfully delivered to the receiver (destination node). The following assumptions are also made:

- (1) Each node has a unique nonzero ID and a mechanism to discover its one-hop neighbors.
- (2) Communication between one-hop neighboring nodes is more reliable compared with multi-hop communication over the error-prone wireless channel.
- (3) Each node has at least k one-hop legitimate neighboring nodes. If a node could not find k neighbors; it may wait for new nodes coming in or roam to a new location for more neighbors.
- (4) Mobility is characterized by an average node moving speed (U_{av}).
- (5) Each node is equipped with some local detection mechanism to identify misbehaving nodes among its one-hop neighborhood.

This last assumption is based on the observation that although intrusion detection in MANETs is generally more difficult than in wired networks [Zha 00], detecting misbehaviors among one-hop neighbors is easier and practical due to the broadcast nature of the wireless transmission [Mar 00].

3.5.2 Adversary models

There are two types of attacks that we are concerned with in this work, these are:

- (1) Denial-of-Service (DoS) attack
- (2) Node break-ins attack

Adversaries may issue DoS attacks from various layers of the network stack ranging from network layer *Smurf* and *Teardrop*, transport layer *TCP flooding* and *SYN flooding*, and various attacks in application layer. For adversaries that seek to compromise networking nodes, we assume that the underlying cryptographic primitives such as RSA are computationally secure.

Occasional break-ins may occur through factors such as insecure OS, software bugs and backdoors, etc. Also, several adversaries may conspire to form a group. For ease of presentation, such an adversary group is denoted by a single adversary. Adversaries are characterized in one of the following two models, as proposed in [Her 95]:

- (1) Model I: During the entire lifetime of the network, the adversary cannot break into or control k or more nodes.
- (2) Model II: Consider the entire lifetime of the network is divided into intervals of length T . During any time interval T , the adversary cannot break into or control k or more nodes.

Although, the adversary cannot break into or control k or more nodes at a particular time, the adversary of model II can choose its victims at each time interval. As time goes on each node in the network can be broken during some time interval.

3.5.3 Intrusion model

At first we briefly discuss what kind of intrusions is allowed in this work. In the worst case, all information, whether public or private, is known to the intruder when a network entity is compromised. The intruder can forge, modify, and delete any information. The intruder can also do bookkeeping to facilitate future break-ins. However, the power of an intruder is set to be limited to make the problem

tractable. Giving infinite power to the intruder simply makes any security design meaningless. Therefore, a more realistic intrusion model needs to be considered in the system.

Authentication is the basic building block for all security services. Fundamentally, it is assumed that each network entity has some information that is unknown to or unforgeable by the intruder. Otherwise, once an entity is broken, there is no way others can differentiate the intruder and the genuine entity. Two specific cases are considered:

- (1) The entity private key will not be exposed for a certain period of time. Thus an entity is able to maintain its security identity by periodically renewing its private key via certificate renewal services.
- (2) The entity ID is not forgeable by the intruder, or the intruder can be detected by intrusion detection mechanisms when it pretends to be the broken entity.

3.6. Simulation Model

3.6.1 Network Simulation

It is generally unfeasible to implement all wireless ad hoc algorithms before valid tests are being performed to evaluate their performance. It is clear that testing such implementations with real hardware is quite hard, in terms of the manpower, time, and resources required to validate the algorithm, and measure its characteristics in desired mobility scenarios. External conditions also can affect the measured performance characteristics. The preferred alternative is to model these algorithms in a detailed simulator and then perform various scenarios to measure their performance for various patterns of node densities, node mobility, radio transmission range, radio environment, size of traffic, etc.

There are a number of simulators that have been developed during the past decade for wireless networks simulators, such as OPNET [Web 1], ns-2 [Web 2],

GloMoSim [Web 3], etc. However, the number of simulators is still growing, and there are 16 out of 63 papers (25.4%) were used a self developed or custom simulators [Kur 06].

The main challenge to simulation is to model the process as close possible to reality; otherwise it could produce performance characteristics entirely different from the ones discovered during actual use. In addition, the simulation study must carefully consider four major factors while conducting credible simulation for MANET research. The simulation study must be [Kur 06]:

- (1) Repeatable and unbiased, the results must not be specific to the scenario used in the experiment.
- (2) Realistic: The scenarios and conditions used to test the experiment must be of a realistic nature.
- (3) Statistically sound: The execution and analysis of the experiment must be based on mathematical principles.

3.6.2 The network simulator (MANSim)

In this work we shall use a newly developed mobile ad hoc (MANET) simulator (MANSim). MANSim is developed to simulate and evaluate the performance of a number of flooding algorithms for MANETs. It is written in C++ language, and it consists of four major modules:

- i. Network module
- ii. Mobility module
- iii. Computational module
- iv. Algorithm module

In order to implement MANSim in evaluating the performance of our TSS algorithm for self-securing MAMETS, we introduce two modifications. One is in the

computational module and the other in the algorithm module. What follows is a description of each of the above modules.

Computational module

Many computational models start a simulation from a single source node positioned at the center of the network area, or from a single source node randomly selected within the network area. The simulation is repeated for S times, i.e., the source node is assumed to transmit S request messages. The results obtained from these simulations are averaged to give average values for the computed parameters. The results obtained reflect the average behavior with regard to this particular source node, but they may not reflect well the average behavior of other nodes within the network.

But, a major feature of MANSim computational module is that it does not randomly pick a node and use it as a fixed source node. Instead, a loop is performed using all nodes within the network as a source node, then the computation for the network parameters is performed sequentially over all nodes, except the source node, as destination nodes. The computed parameters for each source node are averaged over $(n-1)$, and then these averaged values are averaged again over (n) . In other words, the computed parameters are averaged over $(n(n-1))$. In this case, the computed parameters may represent well the average behavior of any of the nodes within the network.

As it has been mentioned earlier, in order to consider node mobility, a simulation time is set. It is divided into a number of time intervals ($nIntv$) that yield a time interval or pause time $\Delta t = T_{sim}/nIntv$, where T_{sim} is the total simulation time. The calculation is repeated, in an outer loop, for $nIntv$, and the results obtained for the computed parameters are averaged over $nIntv$. In general, it has been found that to obtain an adequate network performance, the pause time must be carefully chosen so that the distance traveled by the node, during location update interval, is less than the radio transmission range of the source node. For non-mobile nodes (fixed nodes) $nIntv$ has no effect on the computed parameters and can be set to 1.

Algorithm module

In this module, usually the algorithm is implemented. This module consists of a number of procedures to calculate the computed network parameters. In particular, we develop a procedure to find-out if a source node has succeeded in delivering and receiving a reply that carries the share key of its neighboring nodes. This occurs if the receiving node is within the radio transmission range of the transmitting node and if no error occurs during data transmission due to noise interference, and the source node passes the trust test (i.e., the source node is trustable). Each time a source node i successfully receives a reply, an index $iRec(i)$ is incremented by 1, where i represents the node ID. This index is used to calculate the network parameters. Figure (3.3) outlines the algorithm and the computational modules for TSS algorithm.

Computational Module of the TSS scheme.
Loop over the number of intervals ($m=1, nIntv$) Loop over the number of nodes as source nodes ($i=1, n$) Loop over the number of transmitted request message ($j=1, S$) If (node i successfully authenticated) Then $c = c + 1$; End If Compute $S_R(i)$ for node i as follows: $S_R(i)=c/S$ as given in Eqn. (4). Compute the average value of $S_R(m)$ as follows: $S_R(m) = \sum_{i=1}^n S_R(i) / n$ Compute the average value of S_R as follows: $S_R = \sum_{m=1}^{nIntv} S_R(m) / m$

Figure (3.3) - Computational module of the TSS scheme.

3.7.Performance Measures

The performance of the proposed TSS algorithm for self-securing MANETs is evaluated in terms of a parameter known as the success ratio (S_R). At any time, S_R is calculated as the ratio between the number of nodes that are successfully authenticated or certified access to the network resources (c) and the total number of nodes within the network (n). Thus, S_R can be expressed as:

$$S_R = c/n \quad (3.9)$$

S_R also reflects the probability with which a new arriving node can be successfully authenticated and certified access to the network resources.

In this work we introduce a new parameter to evaluate the performance of the TSS algorithm, which is the sensitivity of S_R to the variation in k . It is referred to as $S(k)$, and it is given by:

$$S(k) = \frac{S_R(k+1) - S_R(k)}{S_R(k)} \times 100 \quad (3.10)$$

Using MANSim, the effect of a number of network parameters on S_R can be investigated, such as:

- (1) Node density (n_d). It is defined as the number of nodes (n) per unit area, i.e., $n_d = n/A$, where A is the network area.
- (2) Node mobility or node speed (u). Nodes are allowed to move with average speed (u_{avg}), maximum speed (u_{max}), or to move with a speed that is randomly selected between zero and some maximum speed (u_{max}), i.e., $u = u_{avg}$, $u = u_{max}$, $u = u_{max} \cdot \alpha$ (α is a random number between 0 and 1), respectively. However, in this work we shall assume that all nodes within the network move with some average speed. A various values of the average speed will be investigated.
- (3) Threshold (k), which is defined as the minimum number of local nodes (one hop legitimate neighboring nodes) that are required to work collaboratively to calculate the system key (SKR).
- (4) Reception probability (p_c). The probability of a request message being successfully received by a destination node located within the radio transmission range of the source node.

- (5) Network trustability factor (T_f). The probability that a destination node, which successfully receives an authentication request message, will positively response to this request.
- (6) Radio transmission range (R). The radio transmission range of the node which is limited and it depends on the radio transmission range of the node and the amount of data transmitted.

3.8 .Practical Implementation of the Proposed TSS Scheme

In this section we discuss the main issues and challenges that are facing the practical implementation of the proposed TSS scheme in real MANETs. These are:

- (1) Obtaining initial certificates. Any new node needs an initial certificate before it can join the network. Moreover, an admitted node has to bear a valid certificate when it requests its certificate to be renewed. The localized certification never creates or issues a brand-new certificate. This policy prevents malicious node to have multiple certificates based by forged or stolen IDs. How to issue initial certificates poses the root of trust problem. A node may issue an initial certificate by an offline authority through external means (e.g., in-person ID). Alternatively, we may use any coalition of k networking nodes to issue an initial certificate via collaborative admission control for this new node. The admission control policy has to be consistent with the robustness of the overall trust model, system model and the adversary models.
- (2) Bootstrapping of the first k nodes. To initialize the very first k nodes, we assume an offline authority who knows the full certificate signing key SKR and the associated polynomial in Eqn. (3.4) of degree $k-1$.
- (3) Parameter k revisited. The design so far assumes each node to have at least k legitimate neighbors. This assumption is critical for certification services to be robust against adversaries. The parameter k also

determines the availability of the services (successful authentication).

- (4) Intrusion detection in ad hoc networks. As presented in the system model (Section 3.4), it is assumed that each node is equipped with some local detection mechanism to identify misbehaving nodes among its one-hop neighborhood. It is believed that as time goes, better local intrusion detection mechanism will be available to serve this purpose.

CHAPTER 4

SIMULATION RESULTS AND DISCUSSIONS

This chapter presents evaluation and analysis of the performance of the proposed threshold secret sharing (TSS) scheme in both noiseless and noisy mobile ad hoc network (MANET) environments, through simulating a number of scenarios using the MANET simulator (MANSim). Four scenarios are simulated to estimate the variation of the authentication success ratio (S_R) with the value of the threshold secret shares (k) and also investigate the effect of various values of input network parameters. Also, for each scenario, the percentage relative change in S_R with k , i.e., the sensitivity $S(k)$ is computed. Definitions of the input and computed parameters were given in Chapter 3.

In particular, each scenario is designed to investigate the effect of a single input parameter (e.g., node density (n), node mobility or node speed (u), radio transmission range (R), and reception probability (p_c)). These scenarios can be summarized as follows:

- (1) Scenario #1: Investigate the effect of node density (n).
- (2) Scenario #2: Investigate the effect of node mobility (u).
- (3) Scenario #3: Investigate the effect of node radio transmission range (R).
- (4) Scenario #4: Investigate the effect of node reception probability (p_c).

The results for these scenarios are discussed and presented in tables and graphs. The rest of this chapter is organized as follows. Sections 4.1 to 4.4 present the simulation results and discussion of the scenarios 1 to 4, respectively.

4.1. Scenario #1: Investigate the Effect of Node Density (n)

Scenario #1 investigates the variation of S_R with k for various values of n . The investigations were carried-out in both noiseless and noisy MANETs environments. For a noiseless MANET environment, the probability of reception (p_c) is equated to 1 (i.e., $p_c=1.0$), while for a noisy environment p_c is equated to 0.8 (i.e., $p_c=0.8$). The input parameters for this scenario are given in Table (4.1).

Table (4.1) Input parameters for Scenario #1.	
Parameters	Values
Geometrical model	Random node distribution
Network area (A)	1000x1000 m
Number of node density (n)	100, 150, 200 nodes.
Transmission radius (R)	150 m
Average node speed (u)	5 m/sec
Simulation time (T_{sim})	1800 sec
Threshold secret shares (k)	1, 3, 5, 7, 9, 11 nodes
Probability of reception (p_c)	Noiseless ($p_c = 1.0$) and noisy ($p_c = 0.8$)
Pause time (\square)	$\square = 0.75 * R / u = 22.5$ sec
Number of runs	20 runs

Table (4.1) shows that the simulation time is 1800 sec and the pause time is 22.5 sec, which means that nodes locations are updated 80 times. Each time, S_R is calculated by dividing the number of nodes that are successfully authenticated by n . A node is considered as successfully authenticated if it establishes a link with k or more nodes from its first-hop neighbors. The values of S_R for all 80 trials are averaged to endow with the simulation S_R . Furthermore, due to the randomness of the process and to enhance the statistics of the results, each simulation is repeated for 20 runs, each run S_R is calculated, and then the average of the S_R values and the associated standard deviation (\square) are calculated. The results for S_R and \square are presented in Table (4.2) and plotted in Figure (4.1).

The main outcomes of this scenario can be summarized as follows:

- (1) As k increases, S_R nonlinearly decreases regardless of the node density for both noiseless and noisy MANETs. This is because when k increases,

- (2) more first-hop neighbors are required to ensure node authentication, a case which can not be satisfied by all nodes all the time due to the random distribution and behavior of the nodes.

Table (4.2) Variations of S_R with k for various values of n .						
k	Noiseless Environment ($p_c=1$)			Noisy Environment ($p_c=0.8$)		
	Node density (n)			Node density (n)		
	100	150	200	100	150	200
1	0.993	0.999	1.000	0.987	0.998	1.000
	(0.008)	(0.003)	(0.000)	(0.011)	(0.004)	(0.002)
3	0.913	0.986	0.997	0.823	0.955	0.987
	(0.028)	(0.010)	(0.005)	(0.040)	(0.017)	(0.009)
5	0.682	0.909	0.976	0.500	0.786	0.923
	(0.044)	(0.024)	(0.013)	(0.057)	(0.032)	(0.020)
7	0.374	0.734	0.907	0.215	0.523	0.775
	(0.073)	(0.041)	(0.020)	(0.073)	(0.045)	(0.037)
9	0.150	0.487	0.779	0.069	0.268	0.565
	(0.069)	(0.054)	(0.033)	(0.051)	(0.049)	(0.047)
11	0.041	0.265	0.592	0.016	0.112	0.344
	(0.036)	(0.065)	(0.044)	(0.025)	(0.050)	(0.052)

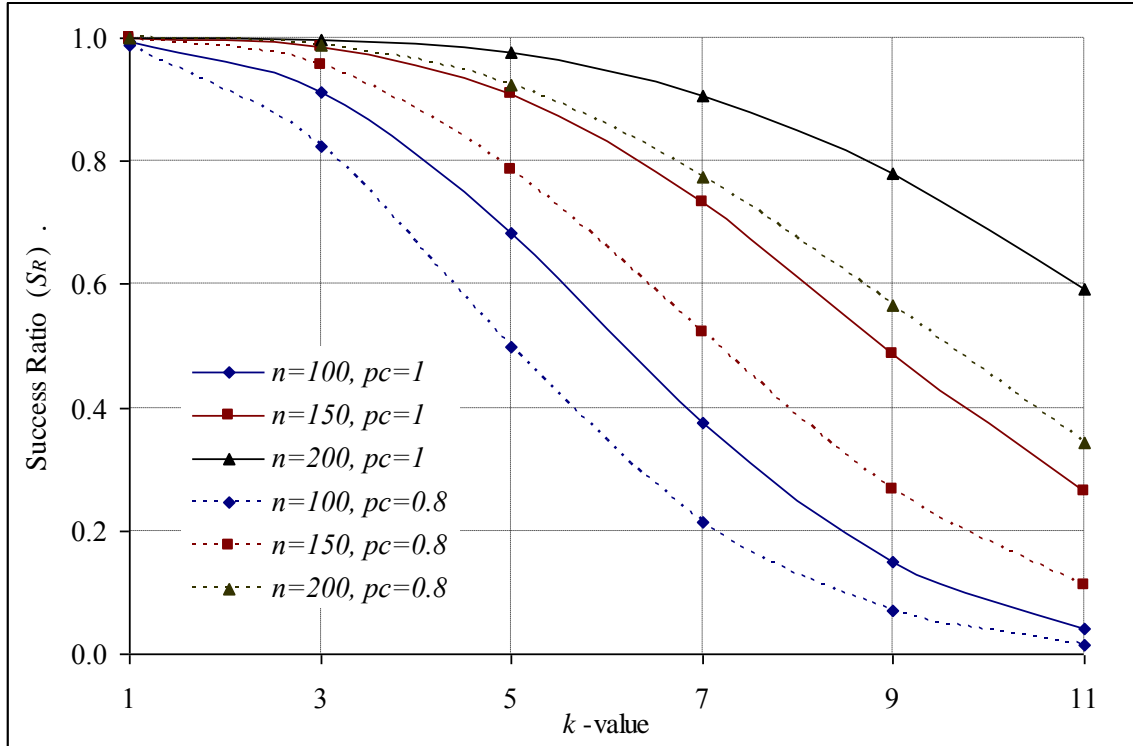


Figure (4.1) - Variation of S_R with k for various values of n and p_c .

- (3) For the same value of k , S_R is directly proportional to n , i.e., as n increases a higher value of S_R can be achieved. Since the node density increases the probability of having neighboring nodes equal to or higher than k nodes is most likely to happen to ensure node authentication.
- (4) For the same node density, when the noise-level increases (i.e., p_c decreases), S_R decreases. This may be explained as follows: When the node whose identity needs to be approved sends an authentication request packet asking for the secret shares of its first-hop neighbors, then due to presence of noise some of these packets may be lost or the requesting node fails to successfully receive its neighbors' replies. For example, if a node physically (distance-wise) has f_1 first-hop neighbors ($f_1 \geq k$), and due to the presence of noise some of the requests or reply packets are lost, and the node practically receives shares from f_2 nodes only ($f_2 < k$), so it can not be authenticated, and the node needs to re-initiate a new authentication request.

In Chapter 3, we introduced a new parameter to evaluate the performance of the TSS scheme, which is the sensitivity of the authentication S_R to the variation of k , namely, $S(k)$. It represents the percentage relative change of S_R at a certain value of k , when k changes from k to $k+1$ as given by Eqn. (3.10). For the S_R values in Table (4.2), the computed values of $S(k)$ are tabulated in Table (4.3) and plotted in Figure (4.2). The negative signs indicate that S_R decreases as k increases.

k	Noiseless Environment ($p_c=1$)			Noisy Environment ($p_c=0.8$)		
	Node density (n)			Node density (n)		
	100	150	200	100	150	200
1	-8.06	-1.30	-0.30	-16.62	-4.31	-1.30
3	-25.30	-7.81	-2.11	-39.25	-17.70	-6.48
5	-45.16	-19.25	-7.07	-57.00	-33.46	-16.03
7	-59.89	-33.65	-14.11	-67.91	-48.76	-27.10
9	-72.67	-45.59	-24.01	-76.81	-58.21	-39.12

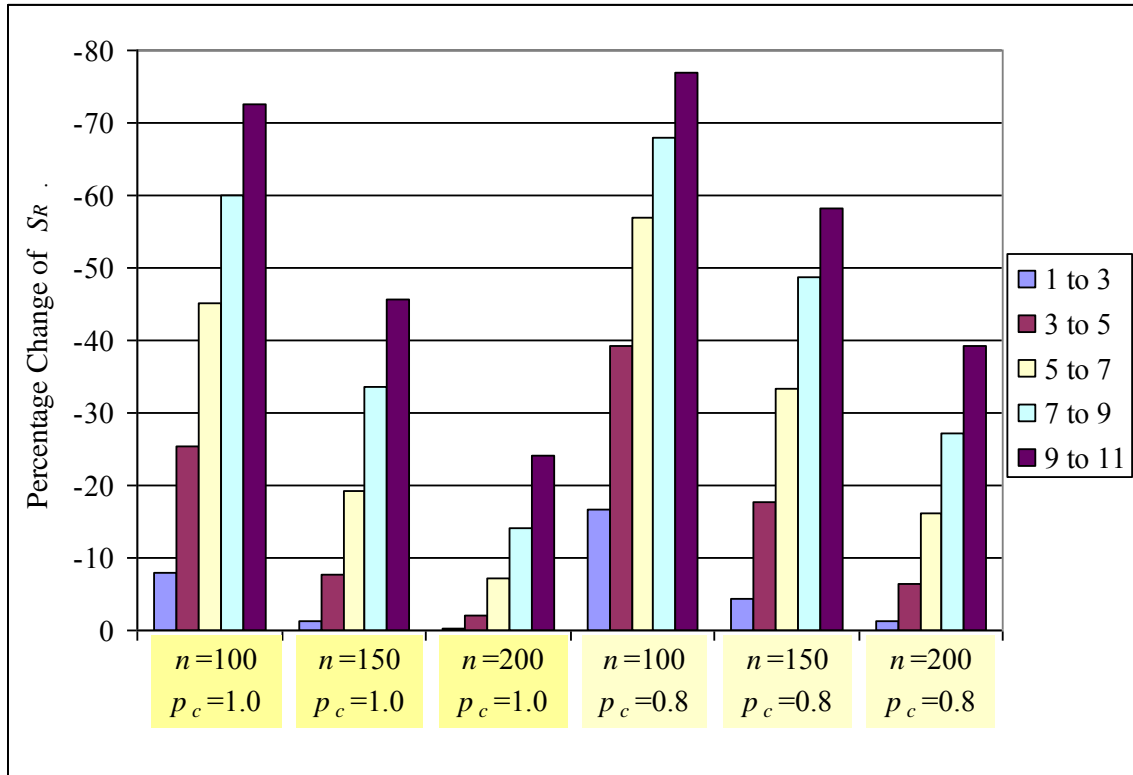


Figure (4.2) - Sensitivity of S_R ($S(k)$) for Scenario #1.

The results in Table (4.3) and Figure (4.2) demonstrate that the percentage change of S_R with k increases as k increases regardless of the nodes densities and in both noiseless and noisy MANETs, which means S_R becomes more sensitive to the change of k at higher k values. However, the sensitivity decreases as n increases. It can also be seen from the results in Table (4.3) and Figure (4.2) that presence of noise significantly increases the sensitivity of the authentication process.

4.2.Scenario #2: Investigate the Effect of Node Mobility (u)

Scenario #2 investigates the variation of S_R with k for various values of u . The investigations are carried-out in both noiseless and noisy MANETs environments. For a noiseless MANET $p_c=1.0$, while for a noisy environment $p_c=0.8$. The input parameters for this scenario are given in Table (4.4).

Table (4.4) Input parameters for Scenario #2.	
Parameters	Values
Geometrical model	Random node distribution
Network area (A)	1000x1000 m
Number of nodes (n)	150 nodes
Transmission radius (R)	150 m
Average node speed (u)	2, 5, 8, 10 m/sec
Simulation time (T_{sim})	1800 sec
Threshold secret shares (k)	1, 3, 5, 7, 9, 11
Probability of reception (p_c)	Noiseless ($p_c = 1.0$) and noisy ($p_c = 0.8$)
Pause time (\square)	$\square = 0.75 * R/u = 56.25, 22.5, 14.0625, 11.25$ sec for $u=2, 5, 8, 10$ m/sec, respectively.
Number of runs	20 runs

The simulations in this scenario are carried-out in the same way explained in Section 4.1 using MANSim simulator. In this scenario four node speeds are

examined, these are 2, 5, 8, and 10 m/sec, which produce different pause times of 56.25, 22.50, 14.0625, and 11.25 sec, respectively. The results for S_R and Δ are listed in Table (4.5) and also plotted in Figure (4.3).

The results show that u has insignificant effects on S_R . The reason for that can be explained as follows: suppose at time (t), the node distribution is as shown in Figure (4.4a), where only three nodes (A, B, and C) can be authenticated out of the four nodes within the network, because they have first-hop neighbors equal to greater than 5 ($k=5$). At time $t+\Delta$, the node distribution is changed where some or all nodes have randomly changed their locations. But still one node (node C) fails to gain access to the network resources because the number of first-hop neighbors it has is less than k nodes, so that it can not be authenticated.

Table (4.5)								
Variations of S_R with k for various values of u .								
k	Noiseless Environment ($p_c=1.0$)				Noisy Environment ($p_c=0.8$)			
	Node speed (u) (m/sec)				Node speed (u) (m/sec)			
	2	5	8	10	2	5	8	10
1	0.999	0.999	0.999	0.999	0.998	0.998	0.998	0.998
	(0.002)	(0.003)	(0.002)	(0.002)	(0.004)	(0.004)	(0.004)	(0.004)
3	0.989	0.986	0.984	0.982	0.951	0.955	0.955	0.953
	(0.009)	(0.010)	(0.012)	(0.012)	(0.019)	(0.017)	(0.017)	(0.018)
5	0.915	0.909	0.907	0.907	0.785	0.786	0.791	0.790
	(0.024)	(0.024)	(0.025)	(0.024)	(0.031)	(0.032)	(0.035)	(0.035)
7	0.738	0.734	0.741	0.745	0.537	0.523	0.528	0.535
	(0.038)	(0.041)	(0.042)	(0.041)	(0.036)	(0.045)	(0.051)	(0.051)
9	0.471	0.487	0.499	0.511	0.283	0.268	0.274	0.283
	(0.056)	(0.054)	(0.053)	(0.058)	(0.051)	(0.049)	(0.060)	(0.060)
11	0.228	0.265	0.273	0.285	0.133	0.112	0.112	0.121
	(0.076)	(0.065)	(0.060)	(0.064)	(0.058)	(0.050)	(0.050)	(0.052)

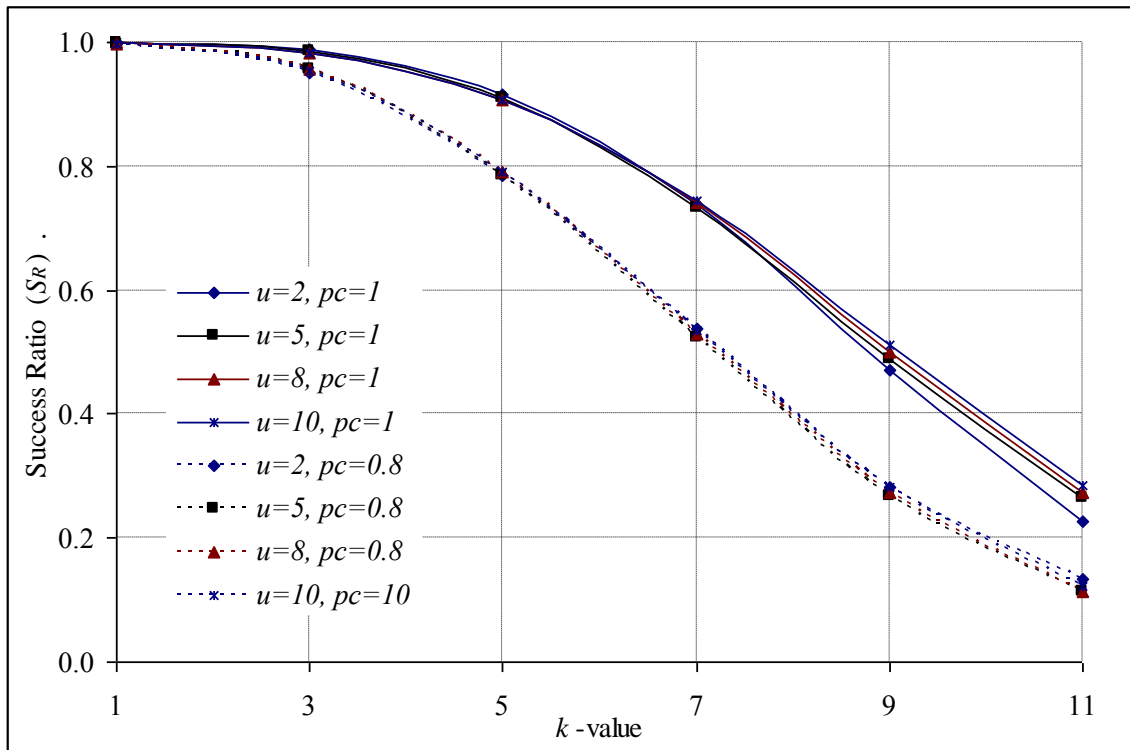


Figure (4.3) - Variation of S_R with k for various u and p_c .

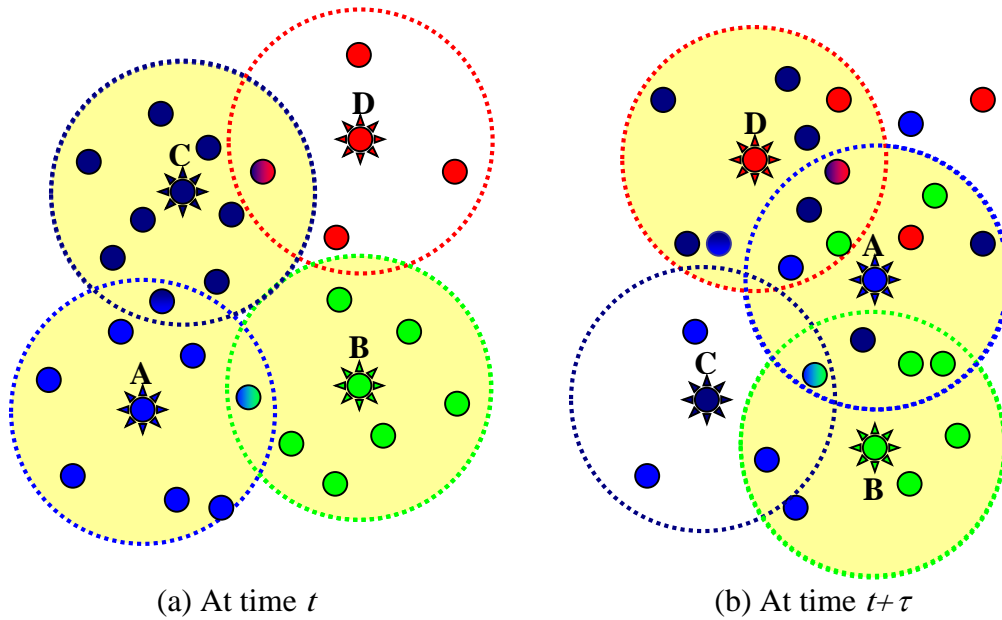


Figure (4.4) - Node distribution at time t and $t+\tau$.

It can also be seen in Figure (4.3) that the same conclusion above is applied to both noiseless and noisy MANETs. But due to the presence of noise some of the first-hop neighbors fail to exchange their secret share with the requesting node so that the requesting node fails to gather k secret shares and it can not be authenticated. Consequently, S_R is less for noisy MANETs as compared to equivalent noiseless MANETs. Since for a certain value of k , S_R is only slightly changed with u , $S(k)$ will not be affected by the variation of u . For $u=5$ m/sec, the values of $S(k)$ were given in Table (4.3). The same values can be taken for $u=2, 8$, and 10 m/sec.

4.3. Scenario #3: Investigate the Effect of Node Radio Transmission Range (R)

Scenario #3 investigates the variation of S_R with k for various values of R in both noiseless and noisy MANETs environments. For a noiseless MANET environment, p_c is equated to 1 (i.e., $p_c=1.0$), while for a noisy environment p_c is taken to be 0.8 (i.e., $p_c=0.8$). The input parameters for this scenario are given in Table (4.6).

Table (4.6) Input parameters for Scenario #3.	
Parameters	Values
Geometrical model	Random node distribution
Network area (A)	1000x1000 m
Number of nodes (n)	150 nodes
Transmission radius (R)	100, 150, 200 m
Average node speed (u)	5 m/sec
Simulation time (T_{sim})	1800 sec
Threshold secret shares (k)	1, 3, 5, 7, 9, 11
Probability of reception (p_c)	Noiseless ($p_c = 1.0$) and noisy ($p_c = 0.8$)
Pause time (\square)	$\square = 0.75 * R / u = 22.5$ sec
Number of runs	20 runs

The results for this scenario are tabulated in Table (4.7) and plotted in Figure (4.5). The results demonstrate that the performance of the TSS scheme is significantly improved with increasing R , where S_R increases as R increases for all values of k . For example, for $k=5$, S_R increases from 38.2% to 99.2% when R increases from 100m to 200m. This is simply because the number of first-hop neighbors (f) most likely increases with increasing R , and if f becomes equal to or greater than k , then more nodes can be authenticated or approved access to the network resources, so that S_R increases.

The relative percentage change of S_R at a certain value of k , which we refer to as the sensitivity $S(k)$ can be deduced from Table (4.7) using Eqn. (3.1).

The computed $S(k)$ values are tabulated in Table (4.8) and plotted in Figure (4.6) for both noiseless and noisy environments. In both environments, $S(k)$ increases as k increases for all values of R . In addition, for a certain k value, $S(k)$ decreases as R increases. This means that S_R becomes less sensitive to the variation of k as R increases. The results also show that in a noisy environment, S_R becomes more sensitive to any variation in k as compared to noiseless environment and same R . For example, for $k=5$ and $R=200$ m, $S(k)$ increases from -7.07% in noiseless environment to -16.03% in noisy environment. Therefore, the k value should be carefully selected even at high radio transmission range, because it may significantly affect the network performance.

Table (4.7)						
Variations of S_R with k for various values of R .						
k	Noiseless Environment ($p_c=1$)			Noisy Environment ($p_c=0.8$)		
	Transmission range (R) m			Transmission range (R) m		
	100	150	200	100	150	200
1	0.980	0.999	1.000	0.959	0.998	1.000
	(0.011)	(0.003)	(0.000)	(0.015)	(0.004)	(0.001)
3	0.766	0.986	0.999	0.642	0.955	0.995
	(0.041)	(0.010)	(0.004)	(0.044)	(0.017)	(0.007)
5	0.382	0.909	0.992	0.264	0.786	0.966
	(0.058)	(0.024)	(0.009)	(0.049)	(0.032)	(0.017)
7	0.119	0.734	0.966	0.067	0.523	0.888
	(0.046)	(0.041)	(0.015)	(0.032)	(0.045)	(0.027)
9	0.025	0.487	0.903	0.009	0.268	0.755
	(0.022)	(0.054)	(0.025)	(0.013)	(0.049)	(0.039)
11	0.004	0.265	0.796	0.001	0.112	0.586
	(0.010)	(0.065)	(0.032)	(0.005)	(0.050)	(0.047)

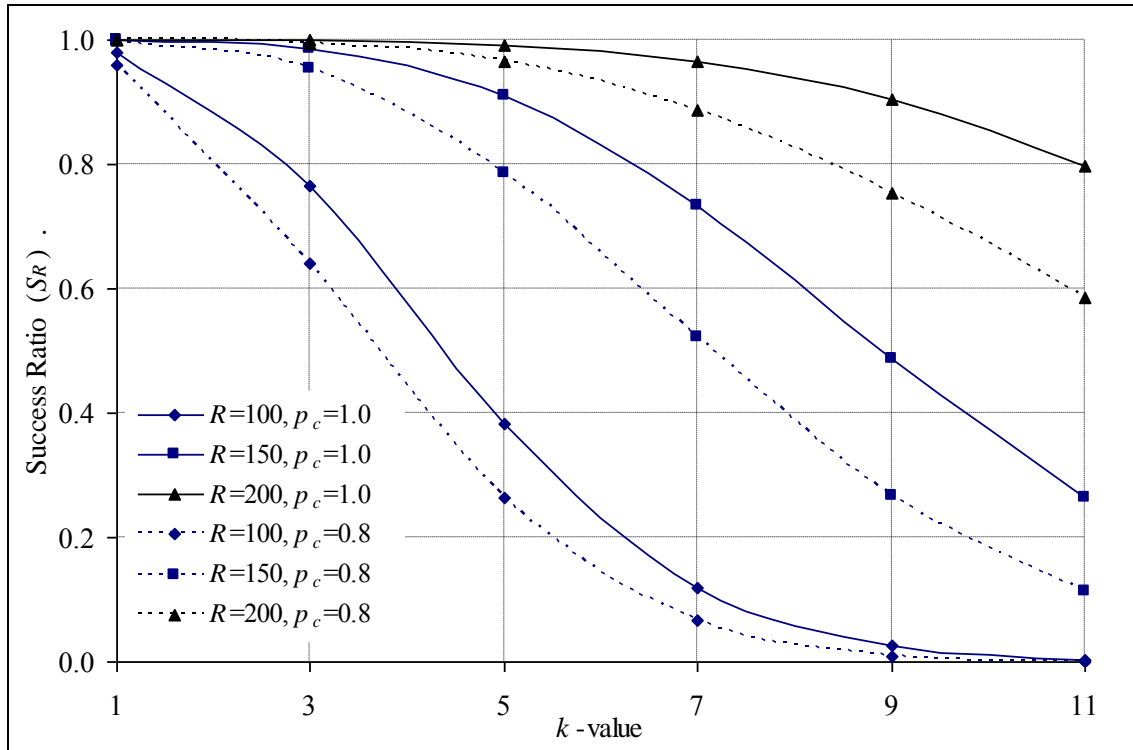


Figure (4.5) -Variation of S_R with k for various values of R and p_c .

Table (4.8)						
Variations of $S(k)$ with k for various values of R .						
k	Noiseless Environment ($p_c=1$)			Noisy Environment ($p_c=0.8$)		
	Radio transmission range (R) m			Radio transmission range (R) m		
	100	150	200	100	150	200
1	-8.06	-1.30	-0.30	-16.62	-4.31	-1.30
3	-25.30	-7.81	-2.11	-39.25	-17.70	-6.48
5	-45.16	-19.25	-7.07	-57.00	-33.46	-16.03
7	-59.89	-33.65	-14.11	-67.91	-48.76	-27.10
9	-72.67	-45.59	-24.01	-76.81	-58.21	-39.12

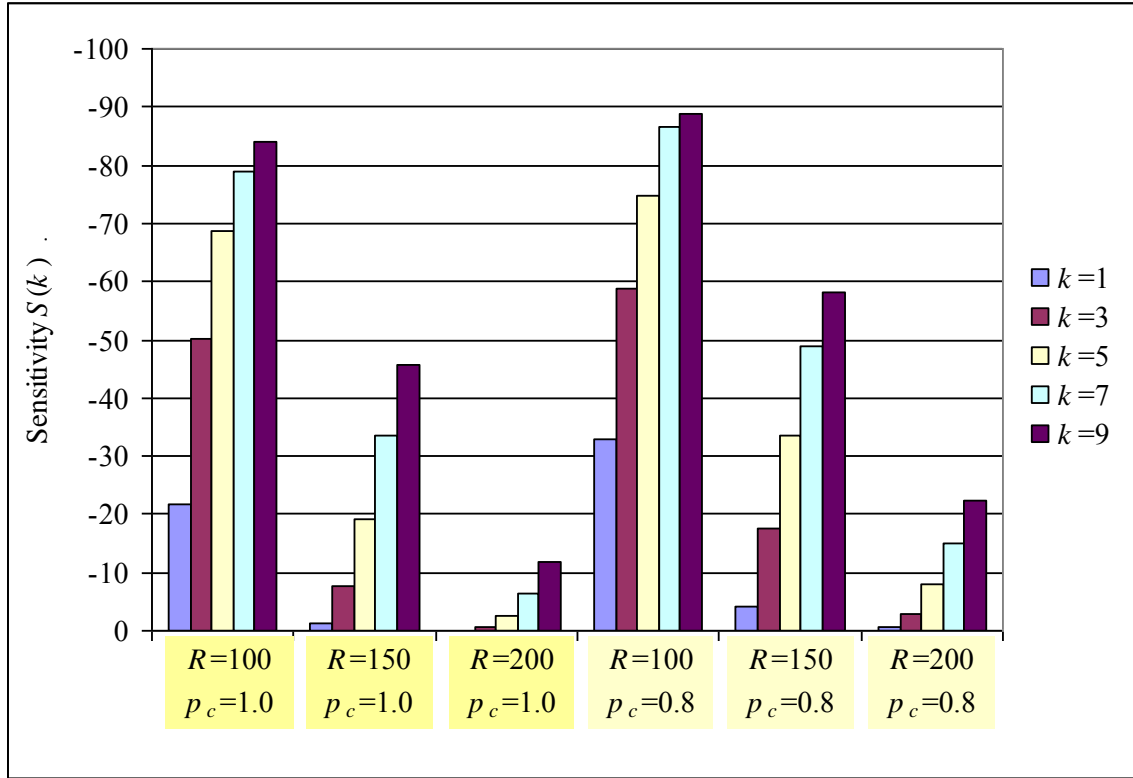


Figure (4.6) - Sensitivity of S_R ($S(k)$) for Scenario #3.

4.4 Scenario #4: Investigate the Effect of Probability of Reception (p_c)

In previous scenarios, all investigations were carried-out for noiseless and noisy MANET environments. For a noisy environment, only a single value of noise-level ($p_c=0.8$) was considered. This may not provide a clear conclusion on the effect of p_c on the performance of the TSS scheme and consequently the performance of the network. Therefore, scenario #4 investigates the variation of S_R with k for a range of p_c values. In fact, the range of p_c values, which were considered, is from 0.5 to 1.0 in step of 0.1. Where $p_c=1.0$ represents a noiseless MANET environment, while the value of $p_c=0.5$ is considered as a high noise-level MANET, where on average half of authentication request and/or reply packets exchanged between the requesting node and its neighbors are lost. The input parameters for this scenario are given in Table (4.9). The results for S_R for this scenario, which

are obtained using MANSim simulator, are presented in Table (4.10) and plotted in Figure (4.7). Table (4.10) also shows the values of \square , which is computed to demonstrate the stability of the computed S_R values.

Table (4.9) Input parameters for Scenario #4.	
Parameters	Values
Geometrical model	Random node distribution
Network area (A)	1000x1000 m
Number of nodes (n)	150 nodes
Transmission radius (R)	150 m
Average node speed (u)	5 m/sec
Simulation time (T_{sim})	1800 sec
Threshold secret shares (k)	1, 3, 5, 7, 9, 11
Probability of reception (p_c)	0.5 to 1.0 in step of 0.1
Pause time (\square)	$\square=0.75*R/u=22.5$ sec
Number of runs	20 runs

Table (4.10) Variations of S_R with k for various values of p_c .						
k	Reception probability (p_c)					
	0.5	0.6	0.7	0.8	0.9	1.0
1	0.986	0.992	0.995	0.998	0.999	0.999
	(0.010)	(0.007)	(0.006)	(0.004)	(0.002)	(0.002)
3	0.787	0.874	0.923	0.955	0.973	0.986
	(0.033)	(0.028)	(0.019)	(0.017)	(0.015)	(0.010)
5	0.417	0.572	0.706	0.786	0.865	0.909
	(0.048)	(0.050)	(0.038)	(0.032)	(0.030)	(0.024)
7	0.145	0.264	0.409	0.523	0.647	0.734
	(0.039)	(0.056)	(0.051)	(0.045)	(0.047)	(0.041)
9	0.036	0.087	0.175	0.268	0.390	0.487

	(0.021)	(0.040)	(0.049)	(0.049)	(0.055)	(0.054)
11	0.004	0.023	0.052	0.112	0.175	0.265
	(0.006)	(0.022)	(0.033)	(0.050)	(0.053)	(0.065)

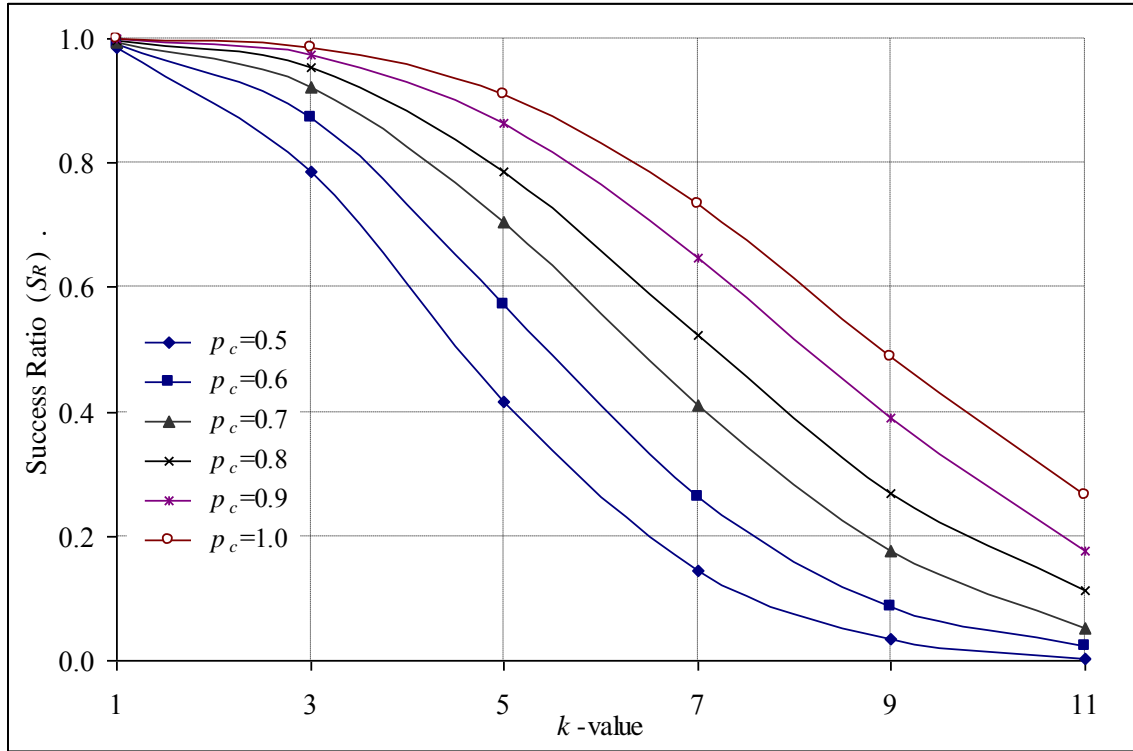


Figure (4.7) - Variation of S_R with k for various values of p_c .

The results demonstrate that S_R inversely proportional to k , where it decreases as k increases for any value of p_c , and for any preset value of k , S_R is directly proportional to p_c , where it increases as p_c increases. This was discussed in Section 4.1. Furthermore, the impacts of the reduction of S_R on the overall performance (communications overhead and delay) of the network were discussed in Chapter 3.

The sensitivity of S_R to the variation of k ($S(k)$) were calculated for various values of p_c using Eqn. (3.1) and presented in Table (4.11). They are also plotted in Figure (4.8). They demonstrate that $S(k)$ varies opposite to S_R , where for any p_c value, $S(k)$ increases as k increases; and for any k value, $S(k)$ decreases as p_c increases.

In conclusion, increasing noise-level not only reducing S_R value but also increasing its sensitivity. Consequently, the noise-level should be carefully considered during the selection of k . Sometimes, it may be necessary to compromise on the security-level to ensure a satisfactory network performance. In addition, presence of noise or increase packet-loss inflicts more reduction on S_R in sparse or low-density networks.

Table (4.11) Variations of $S(k)$ with k for various values of p_c .						
k	Reception probability (p_c)					
	0.5	0.6	0.7	0.8	0.9	1.0
1	-20.2	-11.9	-7.2	-4.3	-2.6	-1.3
3	-47.0	-34.6	-23.5	-17.7	-11.1	-7.8
5	-65.2	-53.8	-42.1	-33.5	-25.2	-19.3
7	-75.2	-67.0	-57.2	-48.8	-39.7	-33.7
9	-88.9	-73.6	-70.3	-58.2	-55.1	-45.6

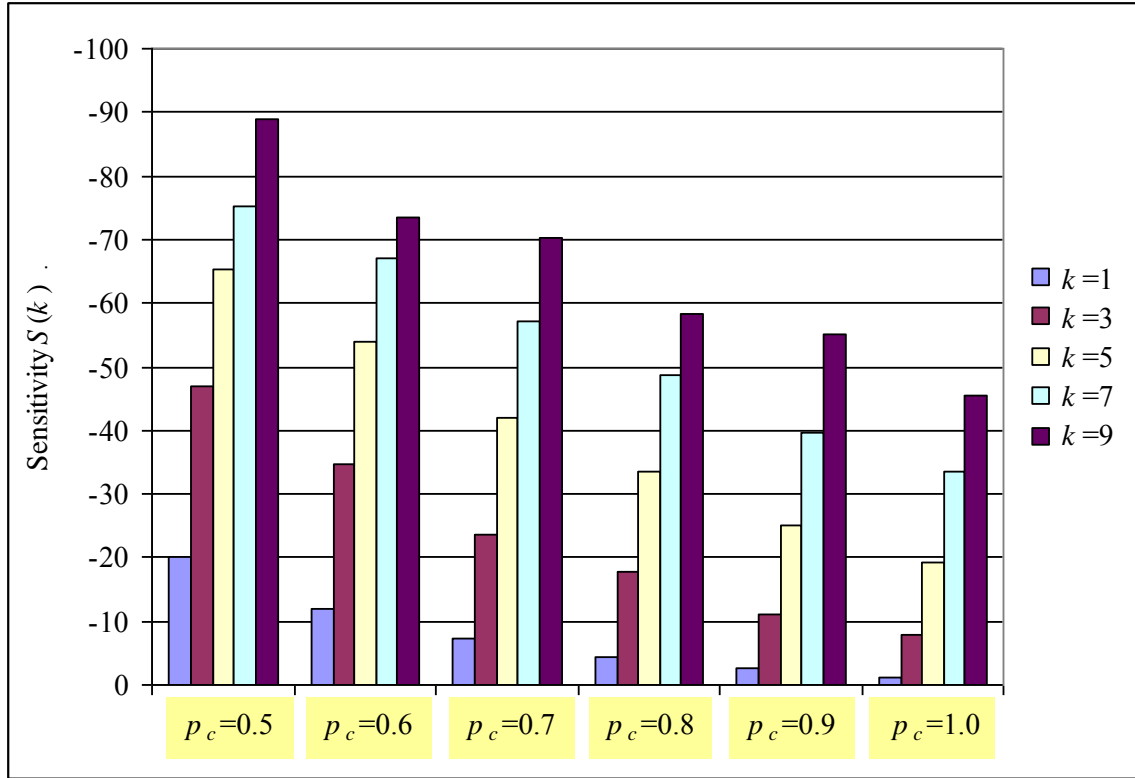


Figure (4.8) - Sensitivity of $S_R(S(k))$ for Scenario #4.

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS FOR FUTURE WORK

5.1. Conclusions

The main conclusions of this work can be summarized as follows:

- (1) The TSS scheme is an efficient and an effective approach that can be used to provide reliable node authentication in MANETs.
- (2) Due to the distribution of the secret shares between the nodes within the network, the scheme provides some satisfying features, such as: the system does not expose to any single point of compromise, single point of denial-of-service (DoS) attack, or single point of failure.
- (3) Due to the localized trust model implemented in this scheme, authentication can be performed in every network neighborhood; this feature is important to authenticate roaming users in a MANET. Furthermore, this solution scales to large network size.
- (4) The authentication security level depends on the threshold secret shares (k) and it is important to select an optimum k that should keep a cost-effective authentication success ratio (S_R), i.e., achieve node authentication with minimum delay and overhead.
- (5) There are a number of network and operating parameters that affect, and should be carefully considered while selecting an appropriate k , such as node density, node speed, node radio transmission ratio, packet-loss rate (noise-level), etc.

This paper investigated the variation of S_R with k for various node densities, node speeds, and packet-loss rates or noise-level expressed in terms of probability of reception. The main conclusions can be summarized as follows:

- (1) Increasing node density has a positive effect on the security-level, since as node density increases higher k value can be selected and still achieving appropriate S_R .
- (2) The node speed has insignificant effects on S_R .
- (3) Increasing noise-level has a negative effect on S_R , therefore, as noise-level increases it is important to reduce k to keep appropriate value for S_R .

5.2.Recommendations for Future Work

The main recommendations for future work may include:

- (1) Evaluating and investigating the variation of the performance of the TSS scheme in terms of other performance metrics, such as: load, thorough, bandwidth utilization, delay, power consumption.
- (2) Evaluating and investigating the variation of the success ratio taking the following into consideration:
 - (i) Allowing nodes from the second-hop neighbors to participate in the authentication process by sending their share keys to the requesting node.
 - (ii) Instead of using a fixed k values, allowing nodes to set k as a location-dependent and/or noise-level-dependent variable. For instance, k may be the majority of each node's neighboring nodes.

References

- [Abd 97] A. Abdul-Rahman. "The PGP trust model". EDI-Forum, the Journal of Electronic Commerce, 1997.
- [Agr 03] D. Agrawal and Q-A. Zeng, **Introduction to Wireless and Mobile Systems**, Cole Publishing, 2003.
- [Akb 08] Rehan Akbani, Turgay Korkmaz, and G.V.S. Raju. "HEAP: A Packet Authentication Scheme for Mobile Ad Hoc Networks". Journal of Ad Hoc Networks, Vol. 6, Issue 7, 1134-1150, 2008.
- [Are 00] A. Aresenault and S. Turner. "Internet X.509 Public Key Infrastructure". Draft-IETF-PIKx-Roadmap-06.txt, 2000.
- [Bal 02] D. Balfanz, D. K. Smetters, P. Stewart, and H. Chi Wong. "Talking to Strangers: Authentication in Ad-Hoc Wireless Networks". Proceedings of Network and Distributed System Security Symposium 2002 (NDSS '02), 2002.
- [Ban 06] M. Bani-Yassein, M. Ould-Khaoua, L. M. Mackenzie, and S. Papanastasiou. "Performance Analysis of Adjusted Probabilistic Broadcasting in Mobile Ad Hoc Networks". International Journal of Wireless Information Networks, 2006.
- [Bra 06] J. Brainard, A. Juels, R. rivest, M. Szydlo, and M. Yung. "Fourth Factor Authentication: Somebody You Know". ACM CCS, 168-178, 2006.
- [Bur 05] M. Burmester and Y. Desmedt. "A Secure and Scalable Group Key Exchange System". Journal of Information Processing Letters, Vol. 94, No. 3, 137-143, 2005.
- [Bur 94] M. Burmester and Y. Desmedt Y. "A Secure and Efficient Conference Key Distribution System". In Advances in Cryptology (EuroCrypt '94),

- Vol. 950 of Lecture Notes in Computer Science, 275-286, 1994.
- [Can 99] R. Canetti, J. Garay, G. Itkis, D. Micciancio, M. Naor, and B. Pinkas. "Multicast security: A taxonomy and some efficient constructions". In INFOCOMM'99, 708–716, March 1999.
- [Cap 03a] S. Capkun, J. P. Hubaux, L. Buttyan. "Mobility Helps Security in Ad Hoc Networks". Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing In ACM MobiHoc, 2003.
- [Cap 03b] Srdjan Capkun, Levente Buttyan, and Jean-Pierre Hubaux. "Self-Organized Public-Key Management for Mobile Ad Hoc Networks". IEEE Transactions on Mobile Computing, Vol. 2, No. 1, 52-64, 2003.
- [Cha 07] Zhenchuan Chai, Zhenfu Cao, and Rongxing Lu. "Threshold Password Authentication against Guessing Attacks in Ad Hoc Networks". Journal of Ad Hoc Networks, Vol. 5, Issue 7, 1046-1054, 2007.
- [Cha 90] George A. Champine, Daniel E. Geer Jr., and William N. Ruh. "Project Athena as a Distributed Computer System". IEEE Computer, Vol. 23, No. 9, 40-51, 1990.
- [Cho 04] Kyu Young Choi, Jung Yeon Hwang, and Dong Hoon Lee. "Efficient ID-Based Group Key Agreement with Bilinear Maps". In Public Key Cryptography (PKC'04), Vol. 2947 of Lecture Notes in Computer Science, 130-144, 2004.
- [Dif 76] W. Diffie and M. Hellman "New Directions in Cryptography". IEEE Transactions on Information Theory, Vol. 22, 644-654, 1976.
- [Epp 02] James F. Epperson. **An Introduction to Numerical Methods and Analysis**. John Wiley and Sons, 2002.
- [For 08] Behrouz. A. Forouzan. **Introduction to Cryptography and Network Security**. Mc-Grew Hill, 2008.

- [For 07] Behrouz. A. Forouzan. **Data Communications and Networking**, McGraw Hill, 4th Edition, 2007.
- [Gar 95] S. Garfinkel. "PGP: Pretty Good Privacy". O'Reilly & Associates Inc., USA, 1995.
- [Hei 99] G. Heine, **GSM Networks: Protocols, Terminology, and Implementation**, Artech House Publishers, 1999.
- [Her 95] A. Herzberg, S. Jarecki, H. Krawczyk, and M. Yung. "Proactive Secret Sharing". Extended Abstract, 1995.
- [Hua 08] Dijiang Huang and Deep Medhi. "A Secure Group Key Management Scheme for Hierarchical Mobile Ad Hoc Networks". Journal of Ad Hoc Networks, Vol. 6, Issue 4, 560-577, 2008.
- [Hub 01] Jean-Pierre Hubaux, Levente Buttyan, and Srdjan Capkun. "The Quest for Security in Mobile Ad Hoc Networks". Proceedings of the 2nd Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001), 146–155, 2001.
- [Hus 08] Shakir M. Hussain and Hussein Al-Bahadili. "A Non-Exchanged Password Scheme for Password-Based Authentication in Client-Server Systems". Science Publications, American Journal of Applied Sciences, Vol. 5, No. 12, 1630-1634, 2008.
- [Hus 06] Shakir M. Hussain and Naim M. Al-Ajlani. "Key Base Random Permutation (KBRP)". Science Publications, Journal of Computer Science, Vol. 2, No. 5, 419-421, 2006.
- [Jar 03] Yousef M. Jaradat, "**Development and Performance Analysis of a Probabilistic Flooding Algorithm in Noisy Mobile Ad Hoc Networks**", MSc.Thesis, Amman Arab University, August 2007.

- [Kim 08] Jongtack Kim and Saewoong Bahk. "Design of Certification Authority Using Secret Redistribution and Multicast Routing in Wireless Mesh Networks". Journal of Computer Networks, in press, corrected proof, available online 10 October 2008.
- [Koh 94] J. T. Kohl, B. C. Neuman, and T. Y. Tso. "The Evolution of the Kerberos Authentication System". In Distributed Open Systems, 78-94. IEEE Computer Society Press, 1994.
- [Koh 93] J. Kohl and B. Neuman. "The Kerberos Network Authentication Service (Version 5)". RFC-1510, 1993.
- [Kom 07] Nikos Komninos, Dimitrios D. Vergados, and Christos Douligeris. "Authentication in a Layered Security Approach for Mobile Ad Hoc Networks". Journal of Computers & Security, Vol. 26, Issue 5, 373-380, 2007.
- [Kom 06] Nikos Komninos, Dimitris Vergados, and Christos Douligeris. "Layered Security Design for Mobile Ad Hoc Networks". Journal of Computers & Security, Vol. 25, Issue 2, 121-130, March 2006.
- [Kur 06] S. Kurkowski, T. Camp, and M. Colagrosso, "MANET Simulation Studies: The Current State and New Simulation Tools", Department of Mathematics and Computer Sciences Colorado School of Mines, Colorado, USA, 2006.
- [Kon 01] Jiejun Kong, Petros Zerfos, Haiyun Luo, Songwu Lu, and Lixia Zhang. "Providing Robust and Ubiquitous Security Support for Mobile Ad-Hoc Networks". Proceedings of the 9th International Conference on Network Protocols (ICNP '01), 2001.

- [Lee 07] Jung-San Lee and Chin-Chen Chang. "Secure Communications for Cluster-Based Ad Hoc Networks Using Node Identities". Journal of Network and Computer Applications, Vol. 30, Issue 4, 1377-1396, 2007.
- [Li 08] Chun-Ta Li, Min-Shiang Hwang, and Yen-Ping Chu. "A Secure and Efficient Communication Scheme with Authenticated Key Establishment and Privacy Preserving for Vehicular Ad Hoc Networks". Journal of Computer Communications, Vol. 31, Issue 12, 2803-2814, 2008.
- [Li 07] Zhenjiang Li, and J.J. Garcia-Luna-Aceves. "Non-Interactive Key Establishment in Mobile Ad Hoc Networks". Journal of Ad Hoc Networks, Vol. 5, Issue 7, 1194-1203, 2007.
- [Lu 05] Bin Lu and U. W. Pooch. "A Lightweight Authentication Protocol for Mobile Ad Hoc Networks". Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC '05), Vol. 2, 546-551, 2005.
- [Luo 02] Haiyun Luo, Petros Zerfos, Jiejun Kong, Songwu Lu, and Lixia Zhang. "Self-Securing Ad Hoc Wireless Networks". Proceedings of the 7th IEEE Symposium on Computers and Communications (ISCC '02), 2002.
- [Mar 00] S. Marti, T. Giuli, K. Lai and M. Baker. "Mitigating Routing Misbehavior in Mobile Ad Hoc Networks". ACM MOBICOM, 2000.
- [Muk 08] Anindo Mukherjee, Anurag Gupta, and Dharma P. Agrawal. "Distributed Key Management for Dynamic Groups in MANETs". Journal of Pervasive and Mobile Computing, Vol. 4, Issue 4, 562-578, 2008.
- [Mur 04] C.S.R. Murthy and B.S. Manoj, **Ad Hoc Wireless Networks: Architectures and Protocols**, Prentice-Hall, 2004.

- [Nar 03] M. Narasimha, G. Tsudik, J. H. Yi. "On the Utility of Distributed Cryptography in P2P and MANETs: the Case of Membership Control". In IEEE ICNP, 2003.
- [Neu 05] B. C. Neuman, T. Yu, S. Hartman, K. Raeburn. "The Kerberos Network Authentication System". Internet RFC 4120, July 2005.
- [Per 02] Adrian Perrig, Ran Canetti, J. D. Tygar, and Dawn Song. "The TESLA Broadcast Authentication Protocol". Journal of CryptoBytes, Vol. 5, Issue 2, 2-13, 2002.
- [Per 00] C. E. Perkins. Ad Hoc Networking. Addison Wesley Professional, 2000.
- [Per 99] R. Perlman. "An Overview of PKI Trust Models". IEEE Network, 38-43, Vol. 13, 1999.
- [Riv 78] R. L. Rivest, A. Shamir and Adleman. "A method of Obtaining Digital Signature and Public Key Cryptosystem". ACM Communication, Vol. 21, No.2, 120-126, 1978.
- [Sha 79] A. Shamir. "How to Share a Secret". Communications of ACM, Vol. 22 Issue 11, 1979.
- [Sta 03] Williams Stallings, Cryptography and Network Security, Prentice-Hall, 4th Edition, 2003.
- [Sun 01] M. Sun W. Feng, and T. Lai, "Location Aided Broadcast in Wireless Ad Hoc Networks", Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM '01), Vol. 5, 2842-2846, San Antonio, Texas, USA, 2001

- [Tan 03] Andrew S. Tanenbaum, **Computer Networks**, Prentice Hall, 4th Edition, 2003.
- [Tse 07] Y. M. Tseng, C. C. Yang, and D. R. Liao. "A Secure Group Communication Protocol for Ad Hoc Wireless Networks". Journal of Advances in Wireless Ad Hoc and Sensor Networks and Mobile Computing, 2007.
- [Var 04] Vijay Varadharajan, Rajan Shankaran, Michael Hitchens "Security for cluster based ad hoc networks". Computer Communications, Vol. 27, Issue 5, 488-501, March 2004.
- [Wan 08] Neng-Wen Wang, Yueh-Min Huang, and Wei-Ming Chen. "A Novel Secure Communication Scheme in Vehicular Ad Hoc Networks". Journal of Computer Communications, Vol. 31, Issue 12, 2827-2837, 2008.
- [Wan 07a] Nen-Chung Wang and Shian-Zhang Fang. "A Hierarchical Key Management Scheme for Secure Group Communications in Mobile Ad Hoc Networks". Journal of Systems and Software, Vol. 80, Issue 10, 1667-1677, 2007.
- [Wan 07b] Guojun Wang, Jie Ouyang, Hsiao-Hwa Chen, Minyi Guo. "Efficient Group Key Management for Multi-Privileged Groups". Journal of Computer Communications, Vol. 30, Issues 11-12, 2497-2509, 2007.
- [Web 1] <http://www.opnet.com>. Available February 2009.
- [Web 2] <http://www.isi.edu/nsnam/ns/index.html>. Available February 2009.
- [Web 3] <http://www.pcl.cs.ucla.edu/projects/glomosim>. Available February 2009.
- [Wu 07] Bing Wu, Jie Wu, Eduardo B. Fernandez, Mohammad Ilyas, and Spyros Magliveras. "Secure and Efficient Key Management in Mobile Ad Hoc Networks". Journal of Network and Computer Applications, Vol. 30, Issue 3, 937-954, 2007.

- [Yeu 08] Chan Yeob Yeun, Kyusuk Han, Duc Liem Vo, and Kwangjo Kim. "Secure Authenticated Group Key Agreement Protocol in the MANET Environment". Journal of Information Security Technical Report, Vol. 13, Issue 3, 158-164, 2008.
- [Zhu 06] Sencun Zhu, Shouhuai Xuy, Sanjeev Setia, and Sushil Jajodia. "LHAP: A Lightweight Network Access Control Protocol for Ad-Hoc Networks". Journal of Computer Networks, Vol.4, Issue 5, 567-585, 2006.
- [Zhu 05] Bo Zhu, Feng Bao, Robert H. Deng, Mohan S. Kankanhalli, and Guilin Wang. "Efficient and Robust Key Management for Large Mobile Ad Hoc Networks". Journal of Computer Networks, Vol. 48, Issue 4, 657-682, 2005.
- [Zhu 04] Sencun Zhu, Sanjeev Setia, Shouhuai Xu, and Sushil Jajodia. "GKMPAN: An Efficient Group Rekeying Scheme for Secure Multicast in Ad-Hoc Networks". Proceedings of the 1st International Conference on Mobile and Ubiquitous Systems (Mobiquitous'04), 2004.
- [Zhu 03] Sencun Zhu, Shouhuai Xu, Sanjeev Setia, and Sushil Jajodia,. "LHAP: A Lightweight Hop-by-Hop Authentication Protocol for Ad-Hoc Networks". In ICDCS 2003 International Workshop on Mobile and Wireless Network (MWN 2003), 2003.
- [Zha 00] Y. Zhang and W. Lee. "Intrusion Detection in Wireless Ad Hoc Networks". Proceedings of the 6th Annual International Conference on Mobile Computing and Networking, ACM MOBICOM, 2000.
- [Zho 99] Lidong Zhou and Zygmunt J. Haas. "Securing Ad Hoc Networks". IEEE Network Magazine, Vol. 13, No. 6, 24-30, 1999.

Arabic Summary

ملخص

إن أسلوب المشاركة الأمنية (TSS) الذي اقترح من قبل شامير قد استخدم على نطاق واسع في تجهيز خدمات التوثيق الموزع للحماية الذاتية في الشبكات اللاسلكية العشوائية المتنقلة. كان هناك العديد من الأبحاث التي تقوم في التحقيق حول أداء هذا الأسلوب في الشبكات اللاسلكية المثالية حيث أظهرت أداء ممتازاً من حيث نسبة نجاح التوثيق، الإتاحة، الموثوقية، والتوقيت.

لكن في الواقع تعاني الشبكات اللاسلكية من خسائر في الحزم نتيجة التشويش والحركة المستمرة لمستخدمي الشبكة، الأمر الذي يؤثر تأثيراً كبيراً في أداء هذا النظام.

إن الهدف الرئيسي من هذا العمل هو تطوير وتقييم أداء التوثيق للحماية الذاتية في الشبكات اللاسلكية العشوائية المتنقلة التي تعاني من خسائر الحزم في مثل (MANETS) وحركة مستخدميها.

الأسلوب المستخدم في هذا البحث يعتمد على مبدأ المشاركة الأمنية (TSS) وقد نفذ باستخدام برنامج (MANSim) الذي تمت برمجته باستخدام لغة ++C.

السمة الرئيسية للتطبيق هو التوثيق من خلال الشبكة المجاورة وهذه الخاصية مهمة لتوثيق المستخدمين المتجولة في الشبكة اللاسلكية.

من أجل تقييم أداء أسلوب (TSS) استخدمت عدد من السيناريوهات للمحاكاة والتي تبين اختلاف نسب النجاح (SR) والذي يعرف عدد محاولات طلب التوثيق الناجحة من بين كل الطلبات خلال وقت معين مع حاملي المفتاح Key لكثافة مستخدمين مختلفة، وسرعة مختلفة، وتعدد مدى نطاق الإرسال وشدة مستوى الضوضاء للشبكة.

تعتبر نتائج هذه السيناريوهات هي في غاية الأهمية لتسهيل كفاءة إدارة الشبكات، وبناء على النتائج التي تم الحصول عليها، تبين أن شدة مستوى الضوضاء في الشبكة تؤدي إلى انخفاض كبير في (SR)، وبالتالي تدهور أداء الشبكة، في حين لم تؤثر أو قد تؤثر بأثر ضئيل حركة مستخدمي الشبكة على مستوى الأداء.

