Key Management for Mobile Ad-hoc Networks

by

CANER BUDAKOGLU
B.Sc, University of Istanbul, 2000

A Thesis Submitted in Partial Fulfillment of the Requirements
for the Degree of

MASTER OF APPLIED SCIENCE

in the Department of Electrical and Computer Engineering

Supervisor: Dr. T. A. Gulliver

ABSTRACT

Designing and implementing any kind of security mechanism requires a secret key, usually known as a cryptographic key, to set up a trust relationship between two or more communicating parties. Key management is the cornerstone of secure communication regardless of the application domain.

In this thesis, a new method of key management is developed by extending the concept of secret sharing within mobile ad-hoc networks, as proposed by Zhou and Haas, to provide for distributed, fault tolerant, hierarchical, robust and reliable security services for these networks. Our new hierarchical approach has two main advantages over the existing solutions: it increases the availability of the security services and helps to categorize security needs for a variety of applications. Simulation results show the effectiveness of our key management scheme in terms of certification success ratio for a variety of mobile ad-hoc network sizes and threshold setups.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

MANET	Mobile Ad-hoc Network
PDA	Personal Digital Assistant
AODV	Ad-hoc On-demand Distance Vector
DSR	Dynamic Source Routing
IEEE	Institute of Electrical and Electronics Engineers
PCA	Partial Certificate Authority
PKI	Public Key Infrastructure
DoS	Denial of Service
MAC	Medium Access Control
IP	Internet Protocol
TTP	Trusted Third Party
GF	Galois Field
DES	Data Encryption Standard
AES	Advanced Encryption Standard
RSA	R. Rives, A. Shamir and L. Adelman
KDC	Key Distribution Center
CA	Certification Authority
PKC	Public Key Cryptosystem
PGP	Pretty Good Privacy
CBR	Constant Bit Rate
ns-2	Network Simulator - 2
ID	Identification
n	The Total Number of Shares

m_t	The Specific Number of Shares ($n \geq m_t$)
t	Threshold Level
$t_{max.}$	Maximum Threshold Level (Highest Security Level)
$t_{min.}$	Minimum Threshold Level (Lowest Security Level)
X_{sender}	Random Secret Integer Generated by a Sender
$X_{receiver}$	Random Secret Integer Generated by a Receiver
p	Prime Number
m	Message
c	Ciphertext
α	Integer
l	Minimum Required Number of the Shares
Z_p	The Set of Integers modulo p
S_t	t^{th} Threshold Level Secret
$< m >_l$	The Signature of m signed by the service private key l.
PS	Partial Signature
$h(m)$	The Hash of the Message(m)

Acknowledgement

I would like to convey my deep appreciation for the continuous support, patience and invaluable assistance I received from my supervisor, Dr. T. Aaron Gulliver. Without the huge amount of fruitful advice and encouragement he provided, this research could have never been completed.

I would like to offer my gratitude to Dr. Amirali Baniasadi, Dr. Sadik Dost and Dr. Bruce M. Kapron for their participation on my committee.

I would further like to acknowledge the support of my colleagues in the Wireless Communication Research Group, namely: Yousry Abdel-Hamid, Dr. Zeljko Blazek, Neil Carson, Richard Chen, William Chow, M. Omar Farooq, Katayoun Farrahi, Massoud Ghassemi, Majid Khabbazian, Wei Li, Yongsheng Shi, Dr. Hao Zhang and Yihai Zhang. Thank you for being such good friends. I value the experiences that you have shared with me from your different cultures.

I would like to express my heartiest thanks to Bridget Minishka for her editing expertise.

I am deeply indebted to the Natural Sciences and Engineering Council of Canada (NSERC) for an industrial postgraduate scholarship and to Sierra Wireless Inc. for its financial support and its sponsorship of our research project.

Most importantly, researching and writing this thesis would not have been possible without the love, understanding and untiring patience of my mother and brother.

Dedication

For my mother and brother

Chapter 1

Introduction

Recently, the demand for more flexible, easy to use and advanced wireless communication technologies has provided opportunities for new networking technologies. Mobile Ad-hoc Networks (MANETs) are an innovative approach to a new form of wireless networking technology. Mobile in the context of this thesis means that nodes in the network may move at differing speeds and directions. Nodes represent mobile devices. They are able to communicate through wireless radio links which have a nominal range of up to 250 meters. Ad-hoc generally means constructed from whatever is immediately available but, for the purpose of this thesis, it means no infrastructure. MANETs are a developing networking technology that require further research to reach their full potential. In particular, MANETs lack solid and robust security mechanisms.

Security is the most crucial implementation issue in many information technologies. Without the appropriate security precautions, critical applications for both commercial and military use, cannot employ any networking technologies.

As wireline technologies are converted to wireless systems, security becomes paramount to the success of the wireless system. Providing security measures for conventional wireline networks is very simple due to well defined cryptographic mechanisms such as public key infrastructure, presence of central support infrastructures and pre-determined topologies. The public key mechanism has superior features over other methods for delivering robust

and reliable security services. The most important advantage of public key infrastructure is that it does not require a secure transmission link between communicating parties. For wireless networks with the presence of a central support structure such as mobile switching centers, base stations, access points and other centralized machinery, we can apply wireline security methodologies, for example, a public key infrastructure.

In the remainder of Chapter 1, we give an overview of a MANET, its routing mechanisms and its applications. This is followed by an introduction to typical security problems in MANETs. After defining fundamental security services and paying special attention to specific security requirements, the main topic of this thesis, key management in MANETs, is introduced. We present a summary of our key management technique before detailing its description in Chapter 3. At the end of this chapter, the organization of the remainder of the thesis is given.

1.1 Mobile Ad-hoc Networks

A mobile ad-hoc network is a collection of mobile routers (and associated hosts) that communicate through mobile links within their radio range. The routers are free to move randomly and organize themselves arbitrarily (they have an arbitrary graph structure) and thus dynamically form a purpose-specific, multi-hop and decentralized radio network. Packet forwarding, routing and other network operations are carried out by the individual nodes. This network definition leads to two new terms: dynamic topology and infrastructure-less network. An illustration of a MANET; including a tablet computer, a cell phone, two personal digital assistants (PDAs) and two laptops is shown in Figure 1.1

In this figure, the tablet computer can communicate directly with the cell phone, PDA-2 and laptop-1 through a wireless link. But, it does not have a direct wireless link with laptop-2 or PDA-1. If the tablet computer wants to communicate with laptop-2 in an ad-

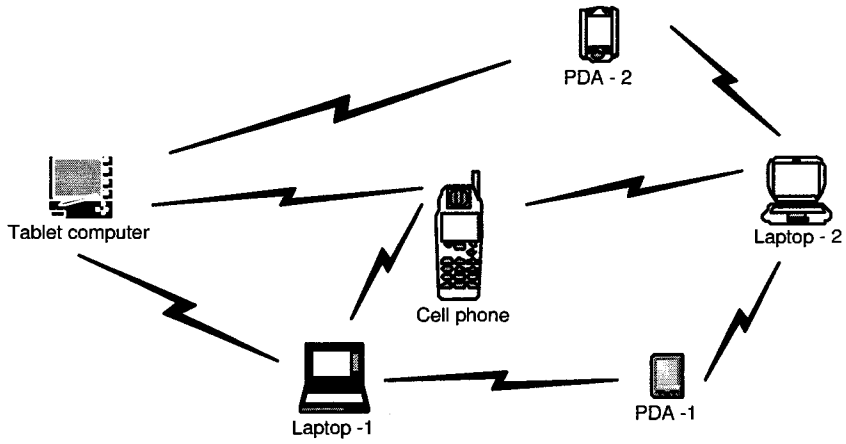

Figure 1.1. *A generic illustration of a MANET.*

hoc fashion, it may use one of the following paths and hops:

1) Tablet computer $\xrightarrow{path1_1}$ PDA-2 $\xrightarrow{path1_2}$ laptop-2 as seen in Figure 1.2
2) Tablet computer $\xrightarrow{path2_1}$ cell phone $\xrightarrow{path2_2}$ laptop-1 $\xrightarrow{path2_3}$ PAD-1 $\xrightarrow{path2_4}$ laptop-2
3) Tablet computer $\xrightarrow{path3_1}$ cell phone $\xrightarrow{path3_2}$ laptop-2
4) Tablet computer $\xrightarrow{path4_1}$ laptop-1 $\xrightarrow{path4_2}$ PDA-1 $\xrightarrow{path4_3}$ laptop-2
5) Tablet computer $\xrightarrow{path5_1}$ laptop-1 $\xrightarrow{path5_2}$ cell phone $\xrightarrow{path5_3}$ laptop-2

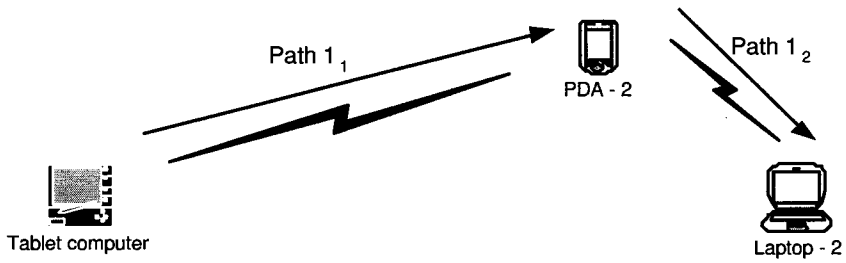

Figure 1.2. *Path 1 and its hops*

A mobile ad-hoc routing protocol tries to determine the best path for communicating parties according to a set of principles. So far, research has been done on a number of routing protocols, such as Ad-hoc On-demand Distance Vector routing protocol (AODV)

and Dynamic Source Routing protocol (DSR). As an example of their differing services, the basic idea behind DSR is source routing ability. The source of a packet decides which route the packet will take to its destination. AODV uses broadcast route discovery which dynamically builds a route by putting the previous hop in each node's routing table along the way. While DSR stores the complete path in its caches, AODV only stores the address of the destination node and the first hop on the path towards the destination in its routing tables.

1.2 Applications of Mobile Ad-hoc Networks

Because of the tremendous flexibility offered by MANETs, new networking technologies such as IEEE 802.11a, b and g and Bluetooth provide demand for successful commercial applications of ad-hoc networks. The applications of mobile ad-hoc networks consist of four main applications: military networks, sensor networks, personal area networks and collaborative networks.

1.2.1 Military Networks

As is commonly the case with most technologies, military operations were the first application of mobile ad-hoc networks because the features of mobile ad-hoc networks match the military's needs. Military applications typically require minimal central infrastructure. In the battlefield, central infrastructure is absent and ubiquitous communication is necessary among military units such as aircraft, tanks, soldiers and other mobile personnel. Communication among units can be established through MANETs. For military purposes, each communication requires the highest level of security. Currently, some countries are testing MANETs in their military operations. Details of military applications can be found at the Tactical Internet [1].

1.2.2 Sensor Networks

Another type of ad-hoc network application is a sensor network. Each node in a sensor network is used to gather information and pass it to a processing center where further analysis and actions can be performed. Sensor networks are different than typical ad-hoc networks. Nodes in a sensor network are usually small in size, extremely limited in power and very low in processing power. Some sensor networks also require a certain level of security depending on the sensitivity of the information.

1.2.3 Personal Area Networks

A personal area network is a network that interconnects a wide variety of mobile devices used by a single person. PDAs, cell phones and laptops are the most common mobile personal area network devices. These devices can easily communicate with each other through wireless radio channels in an ad-hoc fashion. In personal area networks PDAs and laptops may communicate for data transfer and synchronization, while cell phones and laptops may communicate for Internet access. This can be achieved by mobile ad-hoc networking. According to each person's usage, a certain level of security is expected.

1.2.4 Collaborative Networks

The most futuristic mobile ad-hoc network application for consumers is a collaborative network. IEEE802.11a, b, g and hot-spots are successful applications of collaborative networks. The definition of these networks genuinely reflects the main characteristics of mobile ad-hoc networks. The most common examples of these networks are conferences, meetings, coffee shops and restaurants where people come together and wish to communicate with each other for specific purposes through ad-hoc networking. Since there is no need to set-up any infrastructure in advance, ad-hoc networks fit perfectly in this application. Further ad-hoc networks will eliminate the cost of setting up central network support systems. Due to privacy concerns, collaborative networks also need a solid and powerful

security mechanism.

1.2.4.1 Disaster Area Network

Rescue calls and emergency situations are also appropriate applications for MANETs in disaster areas, since there will may be no communication and network infrastructure available for use after the disaster.

1.3 The Security Dilemma in Mobile Ad-hoc Networks

Research and development endeavors into MANETs often focus on finding an ideal routing protocol. If ad-hoc networks are to succeed in the commercial world, the security aspect naturally assumes paramount importance. Security solutions must be devised to prevent attacks that imperil the secure network operation. In 1999, researchers realized that security is a significant implementation issue for ad-hoc networks. Since then, there have been substantial research efforts in both university and industry settings to provide strong and dependable security mechanisms for MANETs.

In conventional wireline networks, the toughest and most infallible security mechanism is the Public Key Infrastructure (PKI) which requires a central trusted third party (TTP) to provide fundamental security services. In Chapter 2, Section 2.8 explains TTP. MANETs continue to face serious security problems due to unique network features such as: mobility, dynamic topology, wireless channel errors, limited energy and cpu speed, limited physical protection of each of the nodes and absence of central infrastructure support. In a dynamic topology, nodes can join and leave the network readily and undetected for some time. We cannot implement the same security structures in MANETs as are successfully utilized in either wireline networks or wireless networks with infrastructure because in MANETs, there are no continuous on-line servers which can act as a TTP.

Establishing a secure environment for a MANET is challenging due to the specific factors mentioned above. Dynamic topology and lack of central infrastructure support are the key features which increase vulnerability and exposure to attacks because well-known security models do not apply to these features. MANETs are vulnerable to several types of attacks including denial of service (DoS), impersonation, eavesdropping and trust attacks.

1.3.1 Classification of Attacks Against Mobile Ad-hoc Networks

Attacks against MANETs can be classified in two ways: external attacks and internal attacks. In turn, these attack classifications can each be subdivided into passive attacks and active attacks as seen in Figure 1.3.

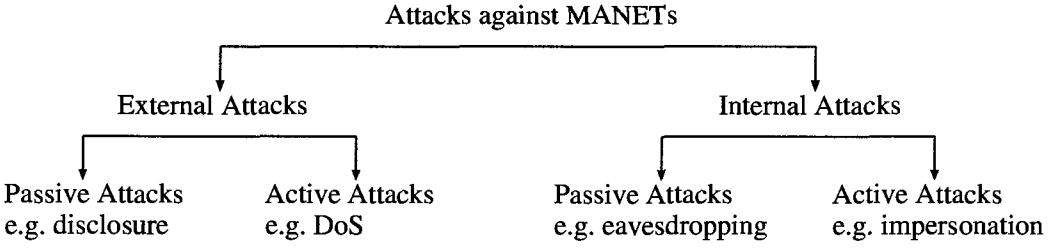

Figure 1.3. *Classification of attacks against MANETs*

- **External Attacks:** These are the most obvious and commonly recognized threat to a MANET. An external attack comes from an adversary node that does not belong to the MANET or share any security context with the MANET. These attacks are targeted to cause congestion, propagate incorrect routing information, prevent security services from working properly or shut the services down completely.
- **Internal Attacks:** It is much more difficult to defend against internal attacks than external attacks since malicious insider nodes already belong to the network as authorized parties and are thus protected by the security mechanisms the MANET offers.

All manner of external attacks are available to an adversary insider node and additional attacks are possible because of the node's participation in network services.

- **Passive Attacks:** These typically involve eavesdropping on transmitted information. Analyzing network traffic and sniffing to compromise keys are two examples of passive attacks.
- **Active Attacks:** These attacks include the replication, modification and deletion of exchanged information. In active attacks, adversaries attempt to change the behavior of the target protocol [2].

1.3.2 Attack Types

The following are the most common attack types:

- **Denial of Service (DoS):** The main target of this attack is minimizing availability of essential network services. The classic way to carry out this type of attack is to overload any centralized resource so that it no longer operates correctly. Radio jamming and battery exhaustion are other types of DoS attacks. In battery exhaustion, due to the ad-hoc network structure, as the target node is flooded, its battery will be run down.

  An adversary may be able to change the routing protocol to operate arbitrarily or perhaps even in the way the adversary wants. If the compromised nodes and the changes to the routing protocol are not detected, the consequences will be serious even though the MANET may seem to operate normally to the other nodes. This kind of invalid working of the MANET, as initiated by malicious nodes, is called a Byzantine failure.
- **Impersonation:** If a weak authentication mechanism is in place, the system will be vulnerable to impersonation attacks such that an external adversary can access network services or gain entrance to the network disguised as one of the trusted nodes. Within a MANET, impersonation can be prevented by strong authentication mecha-

nisms in which a node can trust the origin of the information it has received. Strong authentication mechanisms demand reliable key management services in MANETs.

- **Eavesdropping:** In this case, an adversary watches the MANET traffic and sniffs critical information, such as, specific status details of a node, location of the nodes, private or secret keys, passwords and so on.
- **Trust Attacks:** Several levels of trust can be implemented in a MANET according to organizational privileges which reflect the security, importance, or capabilities of the mobile nodes. To prevent against trust based attacks, MANETs need the access control mechanisms of Authentication, Authorization and Accounting.

1.3.3 Fundamental Security Services

All key management mechanisms should offer differing levels of the following fundamental security services:

- **Confidentiality:** Confidentiality is the protection of the end-to-end transmission of information from active and passive attacks, while guaranteeing that except for the receiver, no one can understand the contents of the transmitted data. This ensures that information is not disclosed to unauthorized entities. Confidentiality is closely related to authenticity, so if authentication is properly applied, confidentiality is a relatively simple process.
- **Authenticity:** Authentication is the proof of identities of communicating parties by ensuring that the origin of the message is verified at the receiver end. Two or more communicating parties are able to match each other's claimed and real identities. Without authentication, an attacker could easily and effectively impersonate a mobile node and gain access to sensitive and classified information.
- **Non-repudiation:** Non-repudiation is somewhat related to authenticity. In non-repudiation, the sender cannot later deny having propagated data to other parts of the network, while the receiver cannot deny reception of the data [3]. This can be

helpful for detecting and isolating compromised nodes. A node that receives an erroneous message can accuse the sender with proof and persuade other mobile nodes about the suspicious node.

- **Integrity:** Integrity ensures that the receiver is able to confirm that the message being transferred has not been modified. A message can be corrupted during transmission by error-prone wireless links or adversary may modify the content of the message.
- **Availability:** With availability, the system security services offered by key management mechanisms will be available to users when expected. These services should be available to mobile nodes at all times.

1.4 Special Security Needs for Mobile Ad-hoc Networks

In addition to fundamental security requirements, MANETs prescribe special security needs such as timeliness, isolation, authorization, low computational complexity, location privacy, anonymity and key management.

- **Timeliness:** Security or routing updates should be delivered in a timely fashion. If the information arrives later than its expected time, it may not be the original message.
- **Isolation:** The protocol is able to identify the malicious nodes. The security system should be designed to be immune to these nodes.
- **Authorization:** An authorized node is issued a non-forgeable credential by the TTP.
- **Low Computational Complexity:** Most mobile devices are battery powered, with limited computational abilities. Nodes cannot afford to carry out complicated computations.
- **Location Privacy:** Sometimes, the information carried in message headers is as valuable as the message itself. In some applications, for example in military networks, location privacy is necessary.

- **Anonymity:** Neither the mobile node nor its system software should expose any information that allows any conclusions about the owner or current user of the node. There should be no direct relationship between the owner of the node and the device or the network identifiers, for instance, MAC address, and Internet Protocol (IP) address.
- **Key Management:** Key management should include the following services: trust mode, crypto system, key creation, storage and distribution. The following section provides more details on key management.

1.5 Key Management for Mobile Ad-hoc Networks

Delivering any kind of security mechanisms using cryptography techniques requires reliable and robust management of cryptographic keys. These keys are the most critical factors in providing a strong security mechanism. In other words, key management is at the heart of every cryptographic system. If an adversary obtains the cryptographic keys in a secure system, overall system security will be compromised.

Key management methods divide into two main categories [4]: centralized key management and distributed key management.

- **Centralized Key Management:** A single central node is responsible for providing most of the security related network services. The node may be a predetermined mobile node or a TTP. Although this technique is simple, effective and the most common solution, its reliance on a single point makes it subject to failure and attack. A single key manager is an easy target for an active adversary who wishes to collapse the network or a passive adversary who wishes to eavesdrop and gather secret information.
- **Distributed Key Management:** Distributed key management requires that a specific number of mobile nodes contribute equally to the generation of a new key. This

technique is based largely on a cryptographic technique known as secret sharing. Essentially, shared keys are a function of a specific number of subkeys from mobile nodes. Because the key is composed of a specific number of mobile nodes, there is no single point of failure. An attack on a node will only prevent that node from joining the network.

Public key infrastructures require TTPs in order to implement key management systems. The main task of a central trusted third party is to validate keys between two or more communicating parties. The central trusted server placed in a physically secure environment should be continuously on-line to offer full-time key management services to communicating parties. If the central trusted server fails, then a secure connection cannot be established. This is also called a single point of failure.

Setting up a trusted server in either wireline networks or wireless networks with central support is not problematic. But in MANETs, promising a continuous on-line server is not pragmatic. Since every node in the network can move freely, resulting in a completely dynamic topology, this is the crucial challenge both for providing robust security and finding optimal routing algorithms. Relying on only one mobile node for assigning trusted third party duties is not a realistic approach in a highly dynamic ad-hoc environment. Various solutions for MANETs to be discussed later have been proposed in the literature [5, 6, 7, 8, 9, 10, 11, 12, 13, 14].

From our point of view, distributing TTP functionality over a specific number of mobile nodes will resolve the security dilemma for MANETs. There are three main reasons to apply distributed security models in MANETs.

1) Centralized approaches are generally not scalable.

2) The trusted third party servers are exposed to single points of compromise and failure.

3) High mobility causes frequent route changes, thus locating and contacting a TTP server in a timely fashion is difficult.

1.6 Overview of Our Key Management Technique

The main rationale behind some proposed models [5, 6, 7, 8, 9, 11, 12] is threshold cryptography. Dividing the secret key into a specific number of shares (n) means that the same secret can be reconstructed with at least the same specific number of shares (m_t) where $n \geq m_t$. (m_t, n) is the notation used in threshold cryptography. There is a $(n - m_t)$ fault tolerance offered in our system. By using threshold cryptography, we are distributing TTP functionality among a specific number of mobile nodes (n) defined as Partial Certificate Authority (PCA) nodes.

We can illustrate secret sharing techniques with the following example. Most critical applications, such as firing nuclear bombs, opening bank vaults and signing important documents require the agreement of multiple parties. In most banks, there is a vault which is opened daily. The bank assigns the duty of opening the vault to nine tellers, as management does not trust the combination to any single teller. Management permits the bank vault to be opened if and only if five or more of the tellers are present. No single teller can open the vault. By solving this problem through combinatorial mathematics as indicated in [15], clearly, for each four tuple of tellers there has to be at least one lock, which cannot be opened by any of them. Whereas, each of the five remaining tellers will have a key for that lock. More than one such lock per four tuple is not needed. So, $\binom{9}{4}$ locks are needed and each teller carries $\binom{9-1}{5}$ keys. After calculating binomial coefficients $\binom{n}{k} = \frac{n!}{k!(n-k)!}$, the minimal solution uses 126 locks and 56 keys per teller.

The solution above is, of course, impractical and becomes exponentially worse if the number of tellers increases. Similarly, the described bank situation is not very realistic. In

contrast, a very real situation occurs when one wants to share sensitive information among a group of people in such a way that only specific privileged coalitions are able to reconstruct the secret information. In our bank example, we need a five-out-of-nine access mechanism and our technique solves this problem.

In a distributed network environment, the solution to providing security services to the network should also be distributed. We improved the proposed threshold based security solution proposed by L. Zhou and Z. Haas [16] by creating a hierarchical structure to the distributed security technique. In our own proposed technique, we improved one of robust and reliable fundamental security services: availability.

We employ differing threshold levels (m_t) in the MANET with a constant number of PCA nodes (n). This will establish easy, efficient and application-specific secure communication between mobile nodes. We can categorize a wide variety of applications according to their security needs.

Our key management technique provides the following desired features for MANETs.

- **Fault Tolerant:** Our key management technique offers reasonable anytime and anywhere service availability for mobile users. Providing such a service for the highly dynamic structure of MANETs is very challenging. Fault tolerance offered in the key management system is $(n - m_t)$.
- **Robust:** The system provides specific (threshold) levels of robustness against break-ins. However, despite the strength of a key management technique, intrusions cannot be eliminated completely in a MANET. Rather than designing a system that is vulnerable to a single point of failure, an attacker needs to break into a number of mobile PCA nodes in order to compromise overall system security.
- **Distributed:** Due to the shortage of transmission range in MANETs and the lack of continuous on-line servers, distributing the trusted third party functionality among a

specific number of mobile PCA nodes provides distributed key management services. This also helps to increase availability.

- **Communication efficient:** Our key management technique works without an assumption about the reliability of the routing protocol; the protocol need only provide basic services such as sending and receiving packets in timely manner. Our design provides communication efficient service because the wireless channel is bandwidth limited and error-prone.
- **Scalable:** As mobile nodes move around, network size changes. Our security technique is able to work with large or small scale networks.
- **Hierarchical:** By assigning differing threshold levels, our technique offers users flexibility in choosing an appropriate security level for a given application. In this hierarchical structure, security needs can be categorized for a variety of applications.

More details about our approach can be found in Chapter 3.

1.7 Thesis Organization

The remainder of this thesis is organized as follows.

Chapter 2 presents the fundamentals of security and necessary background information including public key infrastructure, symmetric and asymmetric cryptosystems, threshold cryptography, an overview of key management and digital signatures.

Chapter 3 discusses and analyzes promising key management solutions [5, 6, 7, 8, 9, 10, 11, 12, 13, 14] that have been proposed in the research literature for MANETs and presents our proposed key management framework. We discuss our system design constraints and how to implement our design in MANETs. After this discussion, simulation results are presented. This chapter also evaluates and compares our proposal to existing solutions.

Chapter 4 concludes the thesis and suggests some topics for future work.

1.8 Summary

In Chapter 1, we identified problems specific to the structure of key management in MANETs. MANETs are communication networks that do not have a pre-existing central infrastructure. The number of nodes in the network can change instantly, so they have a highly dynamic topology. These are the two main reasons why providing security in MANETs is especially difficult.

Chapter 2

Cryptography Basics

Cryptography is the science and art of hiding information. In cryptography, plaintext, which everyone can understand, is converted into ciphertext that hopefully, no one can understand. The process of transforming plaintext into ciphertext is called encipherment or encryption; the reverse process of transforming ciphertext into plaintext is called decipherment or decryption. A cipher, also called a cryptographic algorithm, is the mathematical function used for encryption and decryption [17]. Both the encryption and decryption processes are controlled by a cryptographic key as illustrated in Figure 2.1.

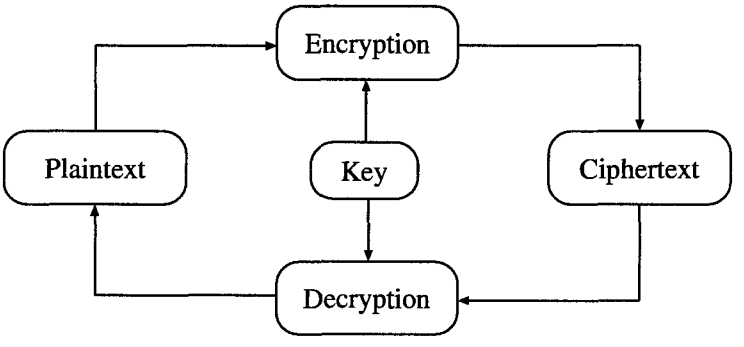

Figure 2.1. *An illustration of encryption and decryption processes.*

In the remainder of Chapter 2, we introduce the two major types of key-based cryptographic algorithms: symmetric key algorithms (private key) and asymmetric key algorithms (public key). After comparing these two methods, we present the Diffie-Helman cryptosystem, the RSA cryptosystem and digital signatures and certificates. We describe

the function of a TTP and explain the necessity for it in networks. We discuss a TTP in a public cryptosystem which is also called a certification authority and provide some insights into key management. We briefly mention key management in public key techniques and the advantages and disadvantages of public key cryptosystems.

2.1 Symmetric Key Algorithms

With a symmetric key algorithm, after plaintext is encrypted, ciphertext is sent through an insecure channel to its receiver. However, before sending any information, the sender and receiver must agree on a key. The decryption key can be easily determined from the encryption key. In most symmetric key algorithms, the encryption and decryption key are the same so these special case algorithms are also called secret key algorithms. As Figure 2.2 illustrates, in a symmetric key algorithm once the receiver obtains the encryption key, the ciphertext can be decrypted. The security of a symmetric key algorithm rests with the key. Accordingly, one of the major issues with a symmetric key algorithm is to find a secure method to exchange the keys.

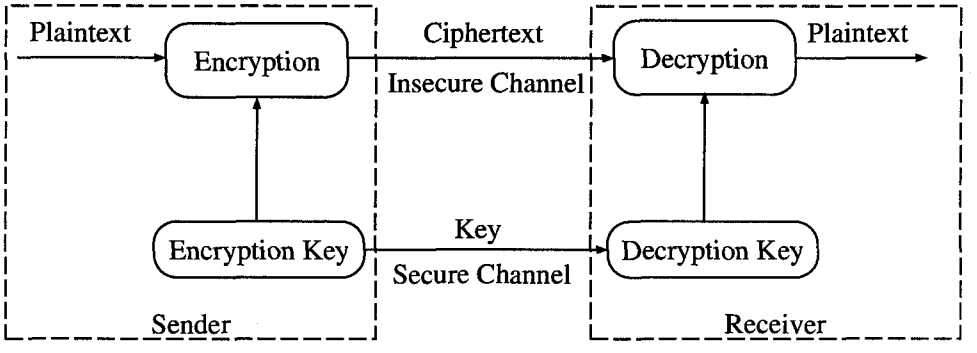

Figure 2.2. *An illustration of a symmetric key algorithm.*

2.2 Asymmetric Key Algorithms

Asymmetric key algorithms are also known as public key algorithms. For the purpose of this thesis, these algorithms are referred to as public key algorithms. Unlike symmetric key algorithms where the sender and receiver share common encryption and decryption keys, a public key algorithm uses a pair of keys, a public key and a private key, which are uniquely associated with each other. Everyone's public key is known in the network and every entity in the network has its own private key as seen in Figure 2.3.

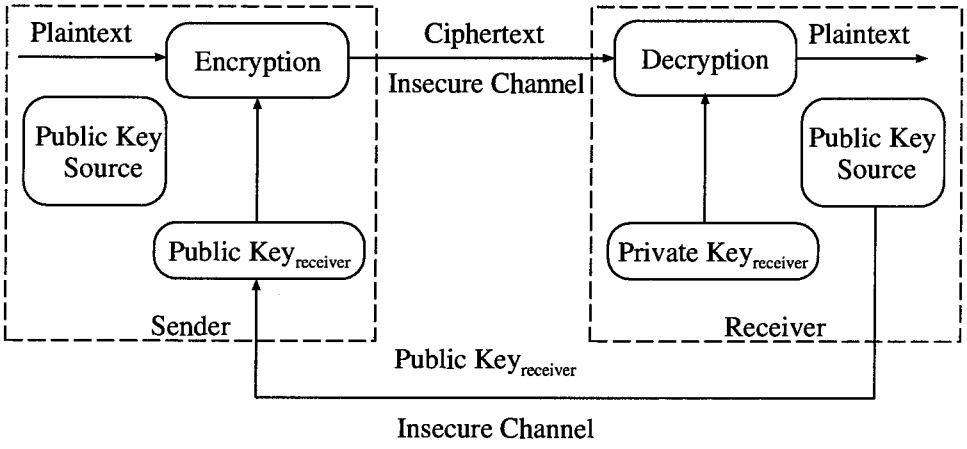

Figure 2.3. *An illustration of a public key algorithm without TTP.*

Public and private key pairs are mathematically related but in such a way that the private key cannot be derived from the public key without additional information. A public key is used for encryption and a private key is used for decryption. To send a message to the receiver that only the receiver can read, the sender uses the receiver's public key ($Public\ Key_{receiver}$) to encrypt the message.

A public key algorithm provides confidentiality because private keys are kept secret. Public key signatures offer authentication, integrity and non-repudiation because the message is also bound by a signature and a private key. The security of any encryption scheme

depends on the key length and the computational effort needed to break a cipher. There are several public key systems that are widely used [17]:

1) Diffie-Hellman Cryptosystem

2) RSA Cryptosystem

2.2.1 An Attack Scenario in a Public Key Cryptosystem without a TTP

The sender can get $Public\ Key_{receiver}$, however it must be authenticated. The sender must ensure that the public key truly belongs to the receiver. As shown in Figure 2.4, an impersonation attack can be made on a public key cryptosystem.

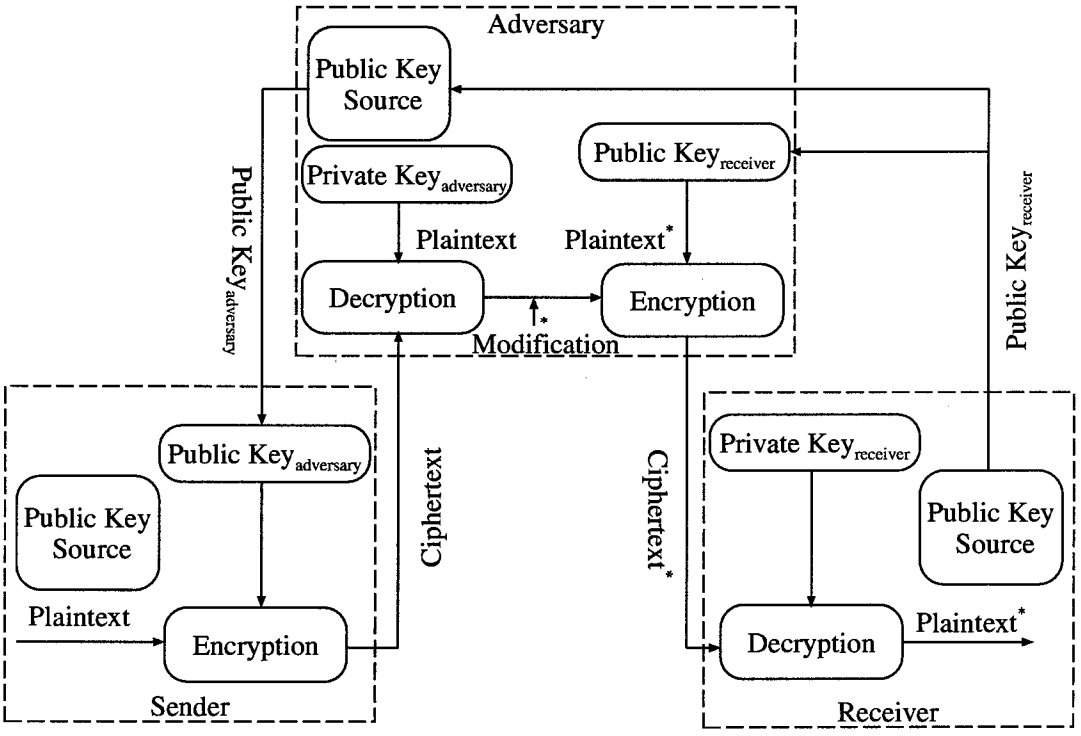

Figure 2.4. *An impersonation attack on a public key system.*

In this scenario, an adversary impersonates the receiver by sending the receiver's modified public key $Public\ Key_{adversary}$ which the sender assumes to be the real public key of the receiver. The adversary intercepts the encrypted message, $Ciphertext$, from the sender to the receiver, decrypts it, into $Plaintext$, with his private key, modifies the message, into $Plaintext^*$, then re-encrypts it into $Ciphertext^*$, with the receiver's public key and sends it to the receiver. The receiver decrypts the $Ciphertext^*$ and gets the message, $Plaintext^*$, as modified by the adversary. This easy impersonation attack highlights the necessity of authenticating public keys to achieve data origin authentication of the public keys themselves. Fortunately, public key techniques offer an elegant solution to this protocol failure problem and this will be explored in Section 2.8. A cryptographic protocol is a set of rules to achieve a specific security objective. Protocol failure occurs when a protocol fails to meet the goals for which it was intended in a such a way that an adversary gains advantage by manipulating the protocol itself [18].

2.3 Comparison of Public and Symmetric Key Cryptography

With symmetric key algorithms, security systems can be designed for high data rates. Symmetric keys are relatively short in size. However, the key must be kept secret to the receiver and the sender. In large networks, there are many key pairs to be managed. Key management requires an unconditional TTP. A TTP is defined as a unconditionally trusted when it is trusted on all matters. For example, it may have access to the secret and private keys of users, as well as be charged with the association of public keys to identifiers [18]. A TTP is defined as a functional trusted when the entity is assumed to be honest and fair but it does not have access to the secret and private keys of users [18].

In a public key algorithm, only the private key must be kept secret; however, authenticity of public keys must be guaranteed. Depending on the network, public/private keys can be

used for a long time, up to several years [18]. Public key algorithms require much greater computational resources than symmetric algorithms. Therefore, public key algorithms are typically used to encrypt a small amount of data (i.e. symmetric encryption keys and digital signatures). Public keys are much larger in size than their symmetric key counterparts. The administration of public keys in a network requires the presence of only a functional TTP as opposed to an unconditional TTP. A digital signature mechanism coming from a symmetric key algorithm typically requires large keys for the public verification function or for the use of a TTP.

In a large network, the number of public keys necessary may be considerably smaller. Throughput rates for the most popular public key algorithms are several orders of magnitude less than symmetric key algorithms [18].

In summary:

- Public key cryptography facilitates efficient digital signatures and key management and
- Symmetric key cryptography is efficient for encryption and some data integrity applications.

2.4 The Diffie-Hellman Cryptosystem

Invented in the 1970s by Whitfield Diffie, Ralph Merkle and Martin Hellman at Stanford University, the Diffie-Hellman cryptosystem is the first public key algorithm [17]. It is based on the difficulty of computing logarithms over the finite field GF (p) which is also called the discrete logarithm problem [17]. A message (m) is converted into ciphertext (c) using the operation given in equation (2.1). Without any additional information about α, it is not feasible to find m as in equation (2.2).

$$c = \alpha^m \bmod p \tag{2.1}$$

$$m = \log_{\alpha} c \bmod p \tag{2.2}$$

If the sender and receiver wish to communicate in secret, they can use the Diffie-Hellman technique to establish a common secret key through the exchange of public messages. This secret key can be used to encrypt a transmission using a conventional cryptosystem, such as the Data Encryption Standard (DES) or the Advanced Encryption Standard (AES).

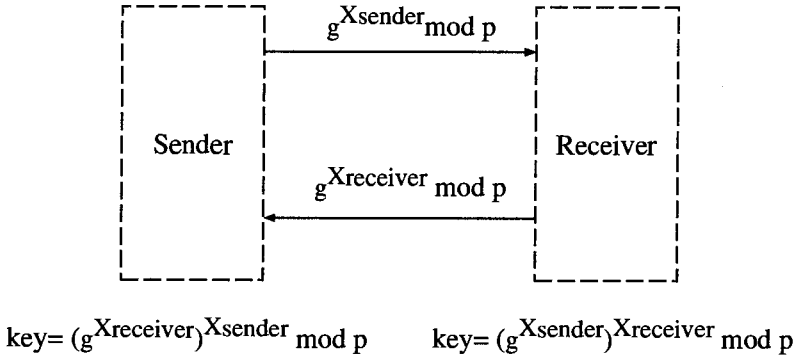

Figure 2.5. *The Diffie-Hellman cryptosystem.*

As illustrated in Figure 2.5, the sender and receiver generate the random secrets, X_{sender} and $X_{receiver}$, respectively. The sender sends $g^{X_{sender}} \ mod \ p$ where p is a prime and g is a generated integer. The receiver sends $g^{X_{receiver}} \ mod \ p$ to the sender and they each calculate the secret key.

2.5 The RSA Cryptosystem

The most famous public key cryptosystem is the RSA cryptosystem, named for its inventors, Ron Rivest, Adi Shamir and Len Adelman of the Massachusetts Institute of Technology. The security of this cryptosystem depends on the difficulty of factoring large prime

numbers. It provides confidentiality, integrity, authentication and non-repudiation services. If a sender and receiver wish to establish a secret key for use with any conventional public cryptosystem, the sender can simply select a secret key of its own and send it to the receiver encrypted with the receiver's public key. The receiver can decrypt the sender's message with the receiver's public key, but that public key cannot be used to decrypt the message, only the receiver's private key will do that. When the receiver obtains the sender's secret key, it can send a message to the sender using its secret key [19].

2.6 Digital Signatures

A digital signature is a data string that can provide authentication, data integrity and non-repudiation. One of the most significant applications of digital signatures is the certification of public keys in large networks. By binding the identity of a user to a public key, other entities can authenticate a public key without assistance from a TTP. The first method used was the RSA signature. The sender computes a hash digest of the message which she encrypts with her private key. A hash digest of the message is also called imprint or digital fingerprint of the message. The sender sends both the message and the encrypted digest which is the signature. The receiver can verify the signature by computing the hash digest of the message he has received and comparing it with the digest he obtains when decrypting the signature using the sender's public key. If the digest matches, the receiver has certainty that the message originated with the sender and that there has been no modification to the message since it was signed. This is illustrated in Figure 2.6.

2.7 Digital Certificates

A digital certificate is a statement issued by a TTP verifying that the $public\ key_x$ belongs to user X. The TTP signs this statement and therefore anyone with the authentic public key of the TTP can verify the certificate and thereafter use $public\ key_x$. An impersonation

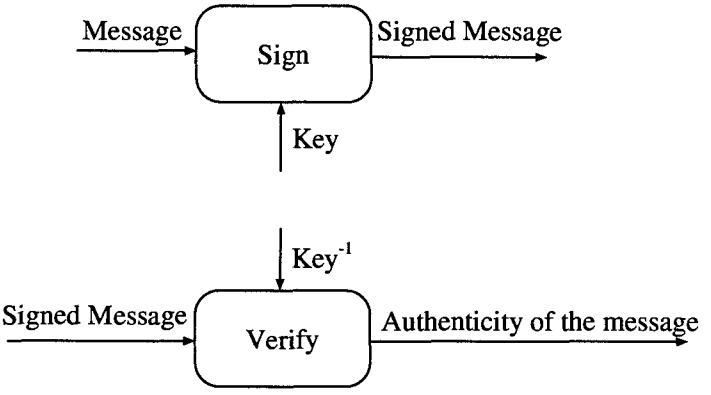

Figure 2.6. *Digital signature.*

attack can be prevented by using digital certificates to verify authenticity of the message.

2.8 Trusted Third Party

The necessity of a TTP for authenticating public keys was discussed in Section 2.2.1. A TTP is a real-time entity trusted by all users of the system. It is designed to solve the public key authentication problem and often provides key management services. TTPs can be divided into three categories [18].

1) In-line TTP: An in-line TTP serves as the real-time means of communication between a sender and a receiver. The sender and receiver communicate through an in-line TTP, as shown in Figure 2.7:

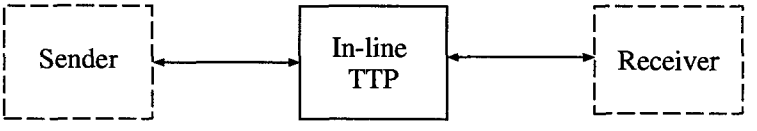

Figure 2.7. *An in-line TTP.*

2) On-line TTP: The sender and receiver communicate with a TTP individually in real time, as shown in Figure 2.8.

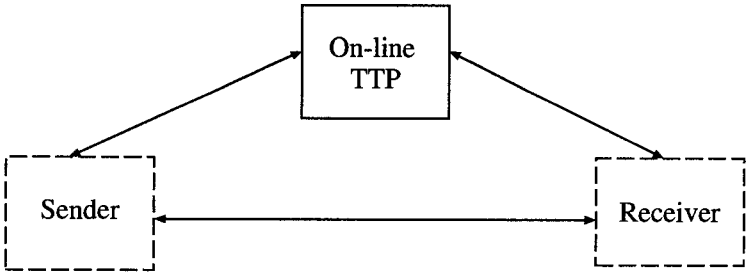

Figure 2.8. *An on-line TTP.*

3) Off-line TTP: An off-line TTP is similar to an on-line TTP. However, with an off-line TTP, communication between the sender and the TTP is performed prior to a protocol run, as shown in Figure 2.9.

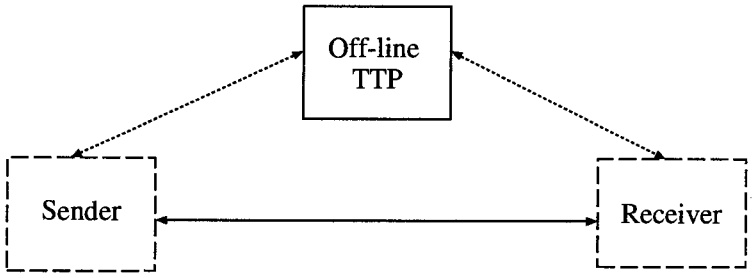

Figure 2.9. *An off-line TTP. [(· · ·) means communication carried out prior to protocol run.]*

Key distribution centers (KDCs) and key translation centers in symmetric key management systems and certification authorities (CAs) in public key management systems are examples of TTPs actively used in networks. KDCs are used to distribute keys to the sender and the receiver. By using TTPs, it is easy to add and remove nodes in a network. Each node needs to store only one long-term secret key, but the TTP is able to read all messages. All communication requires initial interaction with the TTP. The TTP must store all long term keys. If the TTP is compromised, all communication will be insecure.

2.9 Certification Authority in Public Key Infrastructure

A TTP requires several components to offer a complete service: a CA, a name server, a registration authority, a key generator and a certificate directory.

- A CA is responsible for establishing and vouching for the authentication of public keys. This includes linking public keys with names through signed certificates, managing certificate serial numbers and certificate revocation.
- A name server manages the name space of unique user names.
- A registration authority authorizes nodes as a member of a specific security domain.
- A key generator makes public and private key pairs.
- A certificate directory is a database which stores all user certificates.

A public TTP can provide the following basic services: registration, initialization, certification, key update and revocation. TTP services in public key infrastructures are illustrated in Figure 2.10.

One of the most common approaches to authentication and key management in ad-hoc networks uses a TTP. A node that wishes to participate in an ad-hoc network obtains a certificate from a TTP. When two nodes wish to communicate, they must first establish whether the other node has a valid certificate.

A CA is an authority in a network that issues and manages security credentials and public keys for message encryption and decryption. As part of a PKI, a CA checks with a registration authority to verify information provided by the sender of a digital certificate. If the registration authority verifies the sender's information, the CA can issue a certificate.

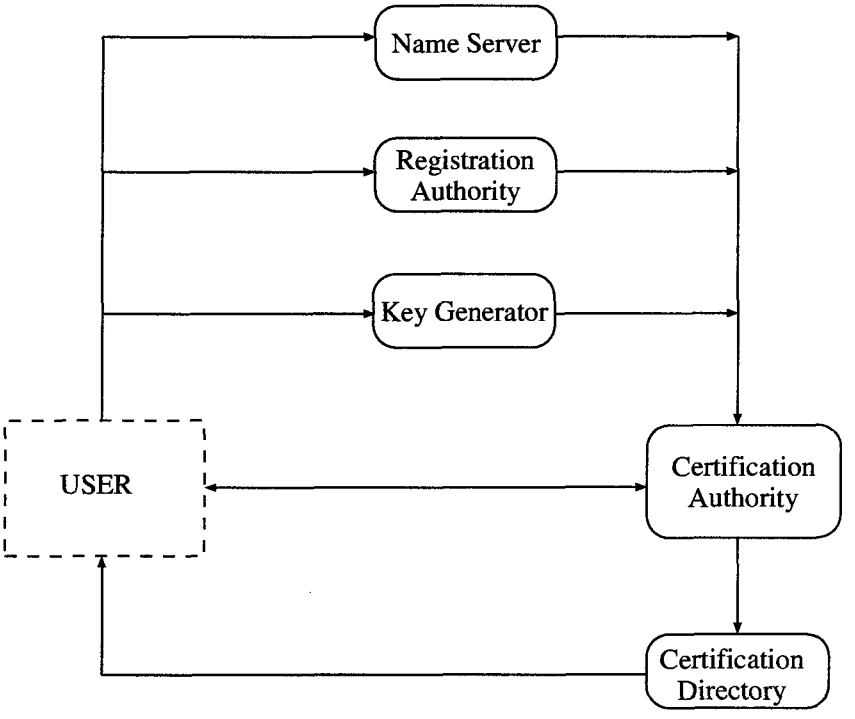

Figure 2.10. *TTP services related to public key certification [18].*

2.10 Hash Algorithm

Another type of cryptographic algorithm, known as a hash algorithm, doesn't use keys. A hash algorithm is a one-way function that maps a message of any size into a fixed size digest. A computationally efficient function that maps binary strings of arbitrary length (hash values), is called a hash function [18]. It should be computationally infeasible for an adversary to calculate the hash value.

Hash algorithms are used with digital signatures for data integrity. A long message is usually hashed using publicly available hash functions and only the hash value is signed. The receiver hashes the message and verifies that the received signature is correct for his hash value.

2.11 Key Management

The goal of key management is to provide secure procedures for handling cryptographic keying material to be used in symmetric or public cryptographic mechanisms. Key management can be defined as generating, storing, distributing, deleting and archiving keys in accordance with a security policy. Key distribution is one of the main problems of key management, namely, the problem of establishing keying material whose origin, integrity and confidentiality can be guaranteed. Another important aim of a key management system is to allow for the authentication of entities by means of keys which involve the registration of users and/or devices [5]. The properties of every key management system are key synchronism, key secrecy, key freshness, forward and backward secrecy, key independence, key authentication and key confirmation [18]. Key management means the control of keying material through the entire lifetime of the keys in order to prevent unauthorized disclosure, modification, substitution, replay and improper use [4].

Key management focuses on communication models for key establishment and use, classification and control of keys based on their intended use, techniques for the distribution of public keys, architectures supporting automated key updates in distributed systems and the roles of TTPs [18].

The purposes of key management have been stated as [18]:

1) Initialize system users within a domain,
2) Generate, distribute and install keying material,
3) Control the use of keying material,
4) Update, revoke and destroy keying material and
5) Store, back-up/recover and archive keying material.

Threats against key management include [17]:

- Compromised confidentiality of keys,
- Compromised authenticity of keys and
- Unauthorized use of keys.

The following are key management services.

- **Registration:** This service maps the user to the user's public key. A registration authority will bind the certificate information, e.g. unique ID and organization, to the public key resulting in a certificate. The information provided by an end user needs to be verified by a registration authority by checking the proof of identity such as a driver's license or ID card. Upon completing verification, the registration authority contacts the CA to request generation of the certificate.
- **Initialization:** In this phase, a public and private key pair will be generated for the entity.
- **Certification:** After receiving a certification request from the registration authority, the CA creates and signs the certificate.
- **Key Update:** A key pair is only valid for a limited time. The key update process requires issuing new key pairs and creating the corresponding certificates.
- **Revocation:** The CA maintains the status of the certificates. If any private key is compromised, the related certification becomes invalid. The CA then revokes the certificate.

Key establishment is any process in which a shared secret key becomes available to two or more parties for subsequent use. Key management is the set of processes and mechanisms which support key establishment and the maintenance of ongoing key relationships between parties, including replacing old keys with new keys as necessary.

For example, consider a network consisting of 5 nodes (users) as illustrated in Figure 2.11. In a symmetric key cryptosystem, there will be $\binom{5}{2} = 10$ possible two party

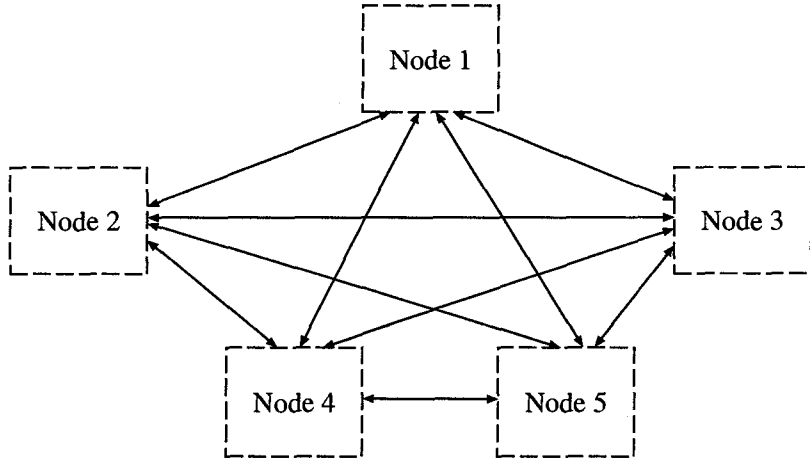

Figure 2.11. *Keying relationships in a simple network.*

communications to exchange 10 key pairs. To become more communication efficient, key management can be applied to a symmetric key cryptosystem.

2.11.1 Key Management in Symmetric Key Cryptosystems

One approach for key management in symmetric key cryptosystems involves the use of a TTP. Symmetric keys are assumed to be distributed over a secure channel prior to the start of communication. When node 3 and node 5 want to communicate, the TTP generates a session key. It will send this key to each entity by encrypting it with their keys , as shown in Figure 2.12.

The advantages of this approach are:

1) It is easy to add and remove entities from a network and

2) Each entity needs to store only one long-term secret key.

The disadvantages are:

1) All communication requires initial interaction with the TTP,

2) The TTP must store long-term secret keys,

3) The TTP has the ability to read all messages and

4) If the TTP is compromised, all communication is insecure.

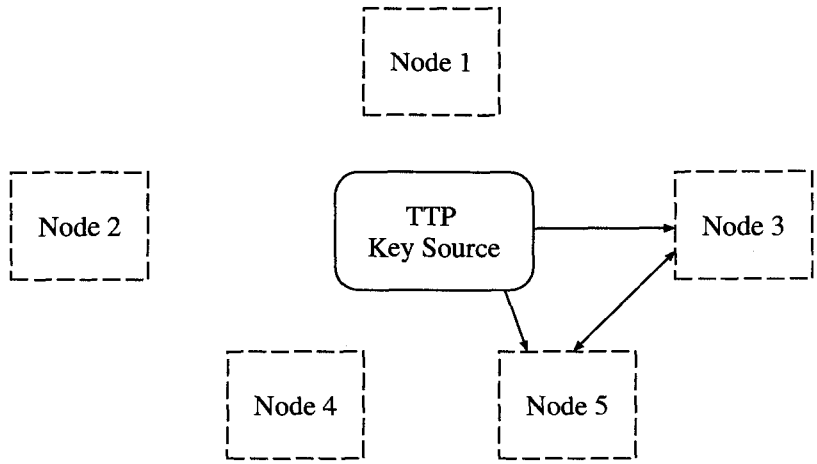

Figure 2.12. *Keying relationships in a simple network with a TTP.*

2.11.2 Key Management in Public Key Cryptosystems

Consider the case when each entity has public/private key pairs. The public key is kept in the central public file repository. The advantages of using a TTP to maintain the integrity of the public file repository are listed below.

1) It prevents an active adversary from making an impersonation attack on the network.

2) The TTP cannot monitor communication. Entities need to trust the TTP only to properly bind identities to public keys.

3) Each communication interaction with the public key file can be eliminated if entities store certificates locally.

The disadvantages are as follows.

1) If the signing key of the TTP is compromised, all communications become insecure.

2) All trust is placed with one entity.

2.12 Advantages and Disadvantages of Public Key Cryptosystems

Public Key Cryptosystems (PKC) have a number of advantages and disadvantages. The advantages include the following:

1) PKCs have simplified key management. To encrypt data, authenticity of public keys is required but not their secrecy.
2) On-line trusted servers are not required. Public-key technique allows a trusted on-line server to be replaced by a trusted off-line server.
3) Extra features are provided by public key techniques. The most notable features are non-repudiation of digital signatures and true data origin authentication [18].

Some of the disadvantages of PKC are as follows:

1) PKCs are computationaly complex.
2) PKCs require longer key size than symmetric key cryptosystems for the same level of security.

2.13 Summary

In this chapter, we examined the background information necessary to understand this thesis. We explained cryptographic basics, including symmetric and public key algorithms. The requirement to use a trusted party in a public key system was described. We also discussed key management and its services.

Chapter 3

Key Management for Mobile Ad-hoc Networks

PKI has been widely used to provide reliable and robust security services in traditional wireline networks. Since we expect MANETs and traditional wireline networks to interact, we chose PKI to also provide security services for MANETs. We use threshold cryptography to distribute TTP functionality over a specific number of PCA nodes as indicated in [5]. Combining these two cryptographic methods provides robust security services for MANETs [5]. We also use a proactive security mechanism to protect MANETs against long-term attacks and a verifiable secret sharing mechanism to prevent DoS in our extended version of Zhou's key management [5].

In the the first part of Chapter 3, we discuss threshold cryptography as proposed by Shamir [20]. After introducing proactive security and verifiable secret sharing, we present key management solutions for MANETs found in the literature including Zhou et al. [5], Hubaux, et al. [13] and Stanjano [10, 21]. We introduce our proposed solution which extends the work of Zhou et al. [5]. We provide simulation results to show the effectiveness of our approach, compare the results to S. Yi and R. Kravets [7, 9, 12] and finally, summarize the chapter.

3.1 Threshold Cryptography

Any application requiring a high level of information security, such as electronic commerce and certification systems, is a potential user of threshold cryptography [22]. The main idea in threshold cryptography, also known as secret sharing, is to divide a secret into a specific number of shares from which the secret can be reconstructed when it is needed. Threshold technique is ideally suited to applications in which a group of mutually suspicious individuals with conflicting interests must cooperate [20]. Because MANETs have these kinds of group characteristics, threshold cryptography provides an ideal security solution. Threshold cryptography can be used to distribute trust among a set of servers and to build a highly available and highly secure key management service in MANETs. The types of security that threshold cryptography can provide include robust confidentiality, integrity, availability and verification.

Threshold cryptography was first introduced by Shamir [20]. His elegant solution for secret sharing introduced new ways to provide highly available key management services in a dynamic MANET environment. In his paper, Shamir showed how to divide message m into n pieces in such a way that m is easily reconstructable from any l pieces where $n \geq l$ and not even complete knowledge of $l-1$ pieces reveals information about m [20]. The notation of threshold cryptography is expressed as (l, n); for example, $(3, 10)$ means that the information is divided into 10 shares and at least 3 shares are needed to reconstruct the information. This information sharing structure can be used in many applications but, most importantly, it can function securely and reliably even when accidental destruction or security breaches expose all but one of the necessary pieces.

Shamir's secret sharing technique is based on polynomial interpolation and works as follows [16, 20, 23, 24]. In general, threshold techniques can be divided into two phases: creating shares and distributing them to nodes (setup) and pooling the shares to rebuild the secret (reconstructing the secret).

3.1.1 Setup

During setup, the TTP T begins with a secret integer $S \geq 0$ that it wishes to distribute among n users [18].

(a) T chooses a prime $p > max(S, n)$, and defines $a_0 = S$.

(b) T selects $l-1$ random, independent coefficients $a_1, a_2, a_3, ..., a_{l-1}, 0 \leq a_j \leq p-1$, defining the random polynomial over Z_p that is the integers modulo p, Z_p, is the set of integers. Addition, multiplication and subtraction in Z_p are performed modulo p

$$f(x) = \sum_{j=0}^{l-1} a_j x^j \tag{3.1}$$

(c) T computes $S_i = f(i) \bmod p, 1 \leq i \leq n$. (or for any n distinct points (i), such that $1 \leq i \leq (p-1)$) and T must securely transfer the share S_i to user P_i, along with a public index i during the initialization.

3.1.2 Reconstructing the Secret

During the pooling of shares, any group of l or more users pool their shares. Their shares provide l distinct points $(x, y) = (i, S_i)$ allowing computation of the coefficients $a_j \leq j \leq l-1$ of $f(x)$ by Lagrange interpolation

$$f(x) = \sum_{i=1}^{l} y_i \left(\prod_{1 \leq j \leq l, j \neq i} \frac{x - x_j}{x_i - x_j} \right) \tag{3.2}$$

In (3.2), $f(0) = a_0 = S$, the secret can be expressed as

$$S = \sum_{i=1}^{l} \left(\prod_{1 \leq j \leq l, j \neq i} \frac{x_j}{x_i - x_j} \right) y_i \tag{3.3}$$

3.1.3 Proactive Security

In threshold techniques, if an attacker has sufficient time, he can compromise l nodes and obtain their shares, thereby allowing him to reconstruct the secret. To prevent against this kind of attack, a proactive security mechanism can be used to update the shares on a regular basis. Accordingly, an attacker must compromise l nodes between the periodic refreshments. Every share has a reasonable lifetime. The new shares are independent of the old shares, except that they define the same secret. As shown in (3.4) to (3.6), the update is achieved by adding a random update polynomial $f_{update}(x)$ where $f_{update}(0) = 0$

$$f_{original}(x) = a_0 + a_1 x + \cdots + a_{l-1}x^{l-1} \tag{3.4}$$

$$f_{update}(x) = 0 + u_1 x + \cdots + u_{l-1}x^{l-1} \tag{3.5}$$

$$\begin{aligned} f_{new}(x) &= f_{original}(x) + f_{update}(x) \\ &= a_0 + (a_1 + u_1)x + \cdots + (a_{l-1} + u_{l-1})x^{l-1} \end{aligned} \tag{3.6}$$

The new shares $S_{i,updated}$ can then be calculated as $f_{new}(i)$, $i = 1, ..., l$. In practice, calculating the shares of the update polynomial and securely distributing them to the respective nodes will be sufficient. Each node can add the update share to its original share to obtain the updated share. If MANETs are used for a long time, a proactive security mechanism should be considered to enhance the existing security structure.

3.1.4 Verifiable Secret Sharing

A verifiable secret sharing mechanism can be used to prevent a DoS attack. For example, during the pooling of shares, if a node wishes to prevent the reconstruction of the secret, it can provide an invalid share to be used for the reconstruction. The Lagrange interpolation will result in a secret, S' which differs from the original secret, S. To prevent this attack, the following steps are applied.

1) Before distributing the shares to the nodes, the initializer who initializes the network broadcasts $\beta^{a_0}, \beta^{a_1}, ..., \beta^{a_{l-1}}$ as witnesses which can be used to verify the coefficients of the sharing polynomial.
2) Now, each node can verify its share by checking $\beta^{S_i} = \beta^{a_0}.(\beta^{a_1})^{ID_i}...(\beta^{a_{l-1}})^{ID_i{}^{l-1}}$.

3.2 Key Management for Mobile Ad-hoc Networks: A Literature Review

In MANETs, there is no guarantee of a fixed central infrastructure or continuous on-line server. Several key management solutions have been proposed [5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 21] in the literature. In [5, 6, 7, 8, 9, 11, 12], threshold cryptography was proposed as a means to distribute trust among a set of servers and to build a highly available and highly secure key management service. Furthermore, these authors employed share refreshing in their key management service to achieve proactive security and to adapt to changes in the network in a scalable way.

In [10, 21], Stajano proposed the resurrecting duckling security policy with key agreement. A device will recognize as its owner the first entity that sends it a secret key. Another approach is a self-organized PKI, as proposed by Hubaux, et al. in [13]. It is similar to Pretty Good Privacy (PGP) in the sense that public key certificates are issued by users. However, as opposed to PGP, users do not rely on certificate directories for the distribution of certificates. Instead, certificates are stored and distributed by the users. PKI has been chosen as our model because of its superiority in distributing keys and in achieving integrity and non-repudiation.

3.3 Key Management with Certificate Chains

Hubaux, et al. [13] presented a self-organized PKI. They replaced the centralized CA by certificate chains. This solution does not require a TTP or any specialized nodes; instead, each user issues its own certificates to other nodes [13]. Users issue certificates when they are able to ensure the identity of the entity they are communicating with. Each node maintains a repository of other node's certificates. When a node wants to validate the certificate of another node, the nodes combine their certificate repositories. The node searches for falsified certificates in the repository. If the node cannot find any falsified certificates, then it validates the certificate of the other node. The node also decides how much trust to place in the certificate. The main drawback of this approach [13], is its initialization time. PGP deals with the problem of distributing public keys in an authentic manner.

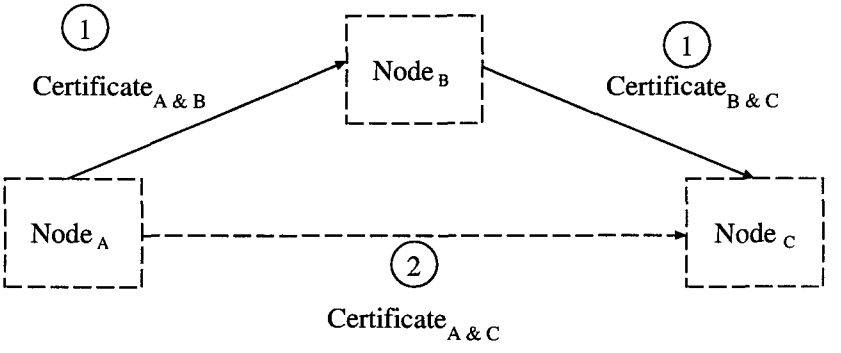

Figure 3.1. *An Illustration of how certificate chains work. [$Node_A$ recognizes $certificate_{B\&C}$ through $node_B$. After that, $node_A$ and $node_C$ can communicate securely.]*

As shown in Figure 3.1, $node_B$ has issued a certificate to $node_C$ stating that *public* key_C really is the public key belonging to $node_C$. $Node_A$ has issued a certificate to $node_B$, indicating that *public* key_B is really the public key belonging to $node_B$. $Node_A$ also trusts $node_B$ not to issue any false certificates, thus $node_A$ will trust any certificates issued by $node_B$. Therefore having $certificate_{A\&B}$ and $certificate_{B\&C}$, $node_A$ can verify that *public* key_C is authentic. Now, $node_A$ can securely communicate with $node_C$ even

though they have never met. As opposed to PGP, public key servers, i.e. certificate directories are not available in a MANET environment and therefore Hubaux, et al.'s solution relies on the users to distribute and store the certificates themselves. Each node stores a number of certificates that have been issued to it. When two nodes wish to authenticate each others public keys, they try to find a certificate chain using only the certificates stored in their combined local certificate repositories.

Although this key management model does not require any form of fixed infrastructure, there are some initial problems before the number of certificates issued reaches an acceptable amount. In the initial stage, the node relies on a user that is trusted to introduce other users and to issue certificates to these other users. This structure relies on a pre-existing trust relationship between the nodes, but does not require trust among all nodes. There must be a gateway or bridge node, as shown in Figure 3.1 where $node_B$ introduces a trust relationship and trust to two other nodes, $node_A$ and $node_C$. If the MANET lacks bridge nodes or the nodes are not all known to each other, there will be problems in setting up secure communications between a sender node and a receiver node, such as $node_A$ and $node_C$, respectively. Managing this type of MANET is not an easy task.

3.4 Key Management with the Resurrecting Duckling

The resurrecting duckling technique was introduced by Stanjano and Andersen [21] and extended in [10]. This technique focuses on MANET devices in the consumer electronics market, with some consideration of medical and industrial devices. It addresses the security services of availability, authenticity, integrity and confidentiality by establishing a master-slave relationship [2]. The authors note the absence of on-line servers and the need for transient security associations because of the transient nature of ad-hoc communication. This technique is transient because the decisions concerning associations between devices can be determined by the master only. Using terminology borrowed from biology, religion

and some cultural sources, the "Resurrecting Duckling" security model was introduced.

When the duckling is born, it recognizes as its mother the first entity that it sees and has physical contact with and sends it a secret key on a secure channel, e.g. physical contact. In ducks, this process is called imprinting. The duckling always obeys its mother (who tells it whom to talk to through an access control list). Like a duckling waiting to hatch, after manufacture, a MANET device is placed in an "pre-birth" state, waiting for a cryptographic master key to be given to it. When the device is used for the first time, the master key given to the device by a physical connection will be accepted as the master authentication key. The owner of the device can issue commands to the "imprinted" device. When the owner decides to cancel the association with the device, a "death" command is issued and the device returns to a pre-birth state, ready for another imprint. If the owner cannot issue a death command for some reason, e.g. loss of master key, the manufacturer should still be able to return the device to its pre-birth state by using a death command.

This technique is secure in the sense that the owner and the device share a common secret. In order to have a certain degree of assurance that the device is real and not a cloned copy with unknown extra functionality, Stanjano and Andersen suggest embedding the device with tamper evident seals. If the device possesses the correct secret key and shows no marks of physical tampering, it should be accepted as the genuine device. Tampering can occur in any subsystem of the device. Although they mentioned public key certificates and signatures, because of resource limitations in 1999 these were not suggested approaches.

The resurrecting duckling technique is an appropriate model for a well defined hierarchy of trust relationships, and is particularly suited to devices without display functions and that have a processor too weak for public key operations. The whole security chain in this technique corresponds to a tree topology formed of hierarchical master-slave relationships.

3.5 Key Management with Threshold Cryptography

Zhou and Haas [5], were one of the first to propose a public key management service for MANETs. A public key cryptosystem can be a solution for MANETs to provide key management services, but the deployment of a PKI requires the existence of a CA which is a TTP responsible for certifying the binding between nodes and public keys. All network nodes know the public key of the CA and trust all certificates signed by the CA's private key. Zhou and Haas distributed CA functionality over a specific number of entities in their key management technique. However, this is not done through simple replication, which would increase the vulnerability of the system, since compromising a single CA would be sufficient for the adversary to control the CA and to compromise the MANET overall. Instead, the authors presented distributed key management where the private key of a trusted service is divided among n servers for enhanced availability and security as shown in Figure 3.2. Trust is distributed among a set of nodes that share key management responsibility.

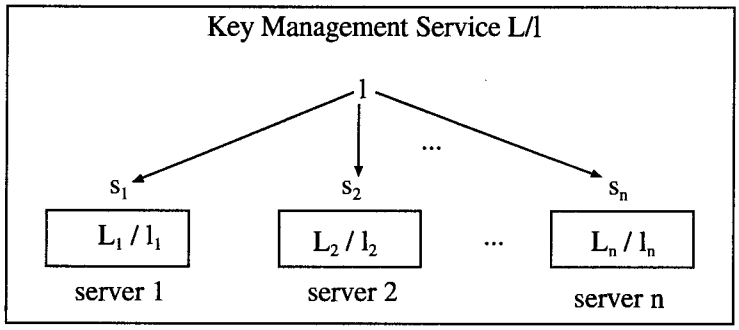

Figure 3.2. *The configuration of a key management service comprising n servers [5].*

At least l out of the n servers pool their shares, s_i, to rebuild a signature with the private key. The actual private key cannot be determined by combining the shares. The signature can be verified by the public key of the service. As mentioned in Section 3.1, a (l, n) threshold cryptographic technique allows n parties to share the ability to perform a cryptographic

operation such as creating a digital signature.

An example is provided in Figure 3.3 for three servers, where L/l is the public/private key pair of the service. The service uses a $(2, 3)$ threshold cryptography technique. Each server i has a share of s_i of the private key l. To calculate the threshold signature on a message m, each server generates a partial signature (m, s_i) and the correct servers S_1 and S_3 forward their partial signatures to a combiner c. Even though the server S_2 fails to submit a partial signature, c is able to generate the signature $< m >_l$ of m signed by the service private key l.

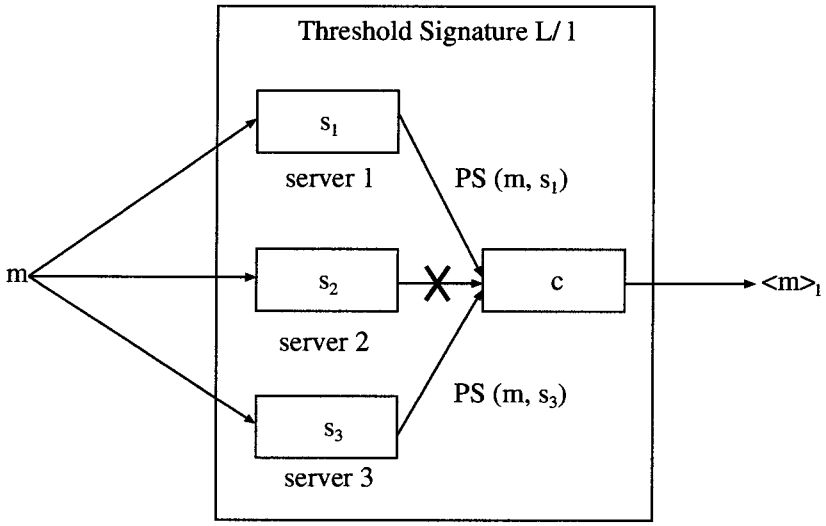

Figure 3.3. *The calculation of a threshold signature using a* $(2, 3)$ *threshold cryptography technique [5].*

In this structure, any l entities can perform the operation together, however $l - 1$ entities cannot reconstruct the signature. If we suppose that, at most, $l - 1$ servers can be compromised at a time, a false signature cannot be created. This will provide $l - 1$ robustness to the security services. Moreover, Zhou, et al.'s key management service also employs proactive security to avoid compromise of l entities from long term attacks. Periodic share updates create new shares of the private key, so that an attacker cannot collect information

about l shares over time. As a result, to compromise the system, all $l-1$ shares have to be compromised within one update period, which can be chosen appropriately short in order to decrease vulnerability. The key management service is also scalable to changes in the number of servers. Before joining the MANET, an entity must first obtain a valid certificate from the initializer off-line. The initializer which initializes the MANET is trusted by all entities.

For high-value transactions PKIs are certainly the best known way to provide a satisfactory and legal security framework [25]. In [5], by distributing the trust of the CA over a specific number of nodes the availability of security services is increased.

3.6 Characteristics of Mobile Ad-hoc Networks

Before designing any routing or security protocol for a network, one should take into account the characteristics of the network. MANETs have the following characteristics:

- **Dynamic Topology**: Nodes in MANETs may move at different speeds and directions and leave and join a MANET at will. Thus, network topology dynamically changes.
- **Limited Bandwidth**: Nodes use wireless radio links for communication. These links have a lower bandwidth and are error-prone.
- **Energy Constrained Nodes**: The biggest obstacle for mobile devices is their batteries. Currently, every mobile device has a limited amount of battery life.
- **Limited Physical Security**: Since each person is responsible for their own device and there is no common physical security for mobile devices, they are more vulnerable to theft and loss than traditional networks.
- **Infrastructureless Networks**: Since MANETs are self-organizing networks, they don't require a fixed infrastructure to establish a network for communicating parties.
- **Fault Tolerant**: A subset of nodes may form a secure subgroup if communication with the rest of the group is lost.

- **Low Computational Complexity**: The computational overhead and memory requirements of the key management system should be minimized.

3.7 Mobile Ad-hoc Networks Security Design Considerations

Standard security solutions are not adequate because they are essentially intended for statically configured systems [2]. MANETs can be classified from differing perspectives. Each application should be considered in the design of the security mechanism. At the physical level, wireless channels offer poor protection to protocol packets and are susceptible to signal interference, jamming, eavesdropping and distortion. When designing a robust and reliable key management technique, one should take into account the following design constraints for mobile ad-hoc networks [26]:

- **Network Origin**: Spontaneous vs. Planned. In planned networks, nodes have a prior relationship i.e. belong to the same company, military etc. In spontaneous networks, nodes have no prior relationship.
- **Network Coverage**: Local vs. Distributed. In local coverage, the nodes are within a certain area and there is the possibility of physical interaction. In distributed coverage, the nodes are distributed over a large area without the possibility of physical interaction.
- **Network Capabilities**: Uniform vs. Diverse. With uniform capabilities, all nodes have approximately the same capabilities in terms of their own features such as cpu, memory, etc. With diverse capabilities, the node capabilities differ significantly, such as a laptop and a PDA.
- **Network Transiency**: Short term vs. Long term. In short term transiency, arbitrary nodes create a MANET temporarily and when finished their communication, they will leave the MANET without gathering information about other nodes. In

long term transiency, nodes are likely to join the same network on regular basis and, accordingly, will save some information about the other nodes for future use.

3.8 Communication Protocol

Our communication protocol is very simple. It consists of two steps: one request broadcast and m_t response unicasts as shown in Figure 3.4. A requester node that wishes to establish secure communication sends a certification request. Then at least m_t PCA nodes acting as distributed CAs must respond to the request with partial certificates.

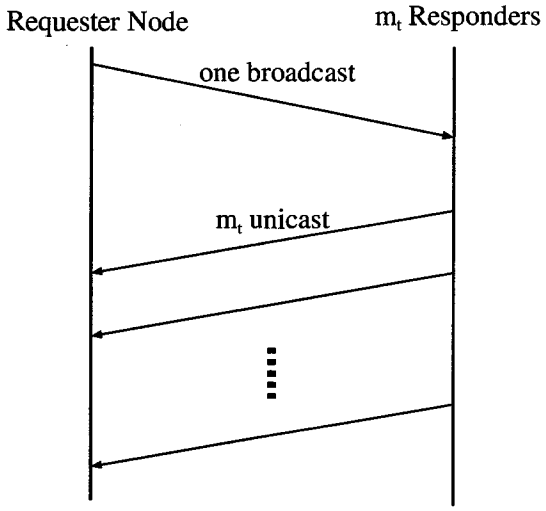

Figure 3.4. *An illustration of the communication protocol*

This operation has minimum requirements to establish reliable wireless communication. As long as m_t PCA nodes respond with partial certificates, the requester node can construct the signature so additional PCA responses may be lost.

3.9 Our Proposed Solution

For the design phase of our key management technique, we make the following assumptions.

1) Each node has a unique nonzero ID.

2) The initializer of the MANET is a trusted party.

3) The network type is planned

Our key management technique is an extension of the work of Zhou and Haas. We use threshold cryptography and threshold function sharing for RSA [27] to distribute the CA functionality among the mobile PCA nodes. Our technique is based on polynomial interpolation and the fact that univariate polynomials $y_t = f_t(x)$ of degree $(m_t - 1)$ are uniquely defined by m_t points (x_{t_i}, y_{t_i}) with distinct x_{t_i} since these define m_t linearly independent equations in m_t unknowns. The mobile trusted initializer begins with secret integers $S_0, ..., S_t \geq 0$ it wishes to distribute among n users. The initializer chooses a prime $p > \max(S_t, n)$ and defines $a_{t_0} = S_t$. The initializer selects $(m_t - 1)$ random, independent coefficients $a_1, ..., a_{m_t-1}$, $0 \leq a_{t_j} \leq (p-1)$, defining the random polynomial over Z_p, as shown below

$$f_t(x) = \sum_{j=0}^{m_t-1} a_{t_j} x^j \tag{3.7}$$

The initializer computes $S_{t_i} = f_t(i) \ mod \ p$, $1 \leq i \leq n$ (or for any n distinct points t_i), $1 \leq i \leq (p-1)$ and securely transfers the share S_{t_i} to user P_{t_i}, along with the public index t_i. If secret sharing is used for reconstructing the secret, the CA's private key gets reconstructed at the requester node. To prevent the reconstruction, threshold digital signatures as proposed by V. Shoup in [28], can be used to generate a digital signature from key pieces without reconstructing the full key at any point. Any node requiring certification service must contact at least m_t PCA nodes with its requests. The contacted PCA nodes each

generate a partial signature from the received data and send it instead of sending their key share. The requester node needs to collect at least m_t such partial signatures to reconstruct the full signature and successfully receive the certification service. PCA nodes enable the requester node to compute $h(m)^{S_t}\ mod\ p$ without learning S_t, the RSA private key. Here, $h(m)$ refers to the hash of the message, m, being signed and p is the RSA modulus. The requester node and PCA nodes perform the Lagrange interpolation implicitly in the exponent of the message as in [27]

$$\begin{aligned} h(m)^{S_t} &= h(m)^{\sum_{i=1}^{m_t} y_i \left(\prod_{1\le j\le m_t, j\neq i} \frac{0-x_j}{x_i-x_j}\right)} \bmod p \\ &= \prod_{i=1}^{m_t} (h(m)^{y_i})^{\left(\prod_{1\le j\le m_t, j\neq i} \frac{0-x_j}{x_i-x_j}\right)} \bmod p \end{aligned} \tag{3.8}$$

After initializer sets up a MANET, nodes are free to move in the network field. If a node wants to set up a secure communication with other parties, it requests partial signutes from PCA nodes. Once it receives the minimum number of partial signatures from PCAs, it can compute the signature without learning CA's actual private key.

Varying m_t between $m_{t_{min}}$ and $m_{t_{max}}$ helps to categorize security needs for a variety of applications and is a hierarchical approach to key management from application perspective. With the hierarchical approach, lowering the minimum threshold level or lowering the minimum required shares increases the availability of security services.

3.10 Simulation Environment

We evaluate our key management framework in terms of its effectiveness from an availability perspective. Effectiveness is measured with a certification success ratio. In our evaluation, every certification request that receives m_t or more certification replies is counted as a successful certification request. The certification success ratio is defined as [12]

$$Success\ Ratio = \frac{Number\ of\ certification\ requests\ received}{Number\ of\ total\ certification\ requests\ sent} \tag{3.9}$$

We used the *ns-2* [29] network simulator to investigate the certification success ratio and the average delay for our key management technique using the Ad-hoc On Demand Vector routing protocol (AODV). A packet transmitted over a single link from a node to another node is called a hop. After that, the packet is forwarded to the next hop. AODV uses a broadcast route discovery approach which dynamically builds a route by putting the previous hop in each node's route table along the way. AODV only stores the address of the destination node and the first hop on the path towards the destination in its routing tables. Details on AODV are given in [30]. In our simulations, we used AODV as a routing protocol because it has the highest successful packet delivery among a number of routing protocols [30]. S. Yi and R. Kravets [7, 9] used AODV when they simulated their proposed key management technique. The wireless model in *ns-2* has a mobile node at its core [29]. A mobile node is able to move within a given topology and is able to transmit and receive signals through a wireless channel.

3.11 Simulation Parameters

The movement models used in the simulations were generated by the *setdest* tool [29]. Similarly, the traffic models were generated using the *cbrgen* tool. These tools are provided by the *ns-2* package. The *ns-2* package provides MANET environment with necessary routing protocols and mobile nodes. The nodes communicate using a Constant Bit Rate (CBR) source with a packet size of 512 KB and a sending rate of 4 packets per second. The CBR generates traffic according to a deterministic rate. Packets are of a constant size. The source-destination node pairs are randomly distributed over the network by the *cbrgen* tool.

The movement models use the random way point model. In all node movement models, the node chooses a destination and moves in a straight line towards it at a speed uniformly distributed between 0 m/s and some maximum speed. This is called the random way point model. Once the node reaches its destination it waits for a pause time before choosing a new random destination.

Simulations were run for six different maximum mobile node speeds: 10, 20, 30, 40, 50 and 60 m/s. For each speed, 8 different simulations were executed with movement models generated for 8 different pause times of 0, 30, 60, 90, 120, 300, 600 and 900 seconds. The length of each simulation was 900 simulated seconds. With 0 second pause time means the node is continuously moving and is defined as having a high mobility. With a 900 second pause time, the nodes move only once. Five networks of varying node size from 10, 20, 30, 40 and 50 node MANETs in the fields of 300m x 300m, 600m x 300m, 900m x 300m, 1200m x 300m and 1500m x 300m respectively were simulated to ensure that this approach is scalable. We ran each simulation 10 times to be consistent in our results. Each non-PCA nodes requested 10 certificates during the each simulation. The time-out threshold was 1 second to receive the minimum number of partial certificates from PCA nodes. In our simulations, the following parameters were used.

1) $(2, 6)$, $(3, 6)$, $(4, 6)$ and $(5, 6)$ threshold setups were used for a network size of 10 nodes,
2) $(2, 12)$, $(4, 12)$, $(6, 12)$, $(8, 12)$ and $(10, 12)$ threshold setups were used for a network size of 20 nodes,
3) $(3, 18)$, $(6, 18)$, $(9, 18)$, $(12, 18)$ and $(15, 18)$ threshold setups were used for a network size of 30 nodes,
4) $(4, 24)$, $(8, 24)$, $(12, 24)$, $(16, 24)$ and $(20, 24)$ threshold setups were used for a network size of 40 nodes, and
5) $(5, 30)$, $(10, 30)$, $(15, 30)$, $(20, 30)$ and $(25, 30)$ threshold setups were used for a network size of 50 nodes.

A specific number of PCA mobile nodes, ($m_{t_{max}}$) act as a mobile CA with the minimum required threshold level ($m_{t_{min}}$). According to security needs, m_t can be adjusted up to ($m_{t_{max}}$). For example, a 30 node MANET has a $(3, 18)$ minimum threshold level, ($m_{t_{min}}$) and a $(15, 18)$ maximum threshold level, ($m_{t_{max}}$). There are an additional 3 levels between the maximum and minimum threshold levels. In each network model, all nodes move according to the parameters selected above. Our overall variable parameters can be seen in Table 3.1 and fixed simulation parameters can be seen in Table 3.3. To see the affect of node density in the network field, we ran simulations with varying node density for a 50 node MANET as seen in Table 3.2.

Variable Simulation Parameters	MANET size (number of nodes)				
	10	20	30	40	50
Network Field (m^2)	300 x 300	600 x 300	900 x 300	1200 x 300	1500 x 300
Threshold Level 1	-	(2, 12)	(3, 18)	(4, 24)	(5, 30)
Threshold Level 2	(2, 6)	(4, 12)	(6, 18)	(8, 24)	(10, 30)
Threshold Level 3	(3, 6)	(6, 12)	(9, 18)	(12, 24)	(15, 30)
Threshold Level 4	(4, 6)	(8, 12)	(12, 18)	(16, 24)	(20, 30)
Threshold Level 5	(5, 6)	(10, 12)	(15, 18)	(20, 24)	(25, 30)

Table 3.1. *Overall variable simulation parameters with respect to MANET size. Threshold level 1 (min.) to Threshold level 5 (max.).*

Simulation Parameters		
Threshold Level 1	(2, 10)	(4, 20)
Threshold Level 2	(4, 10)	(8, 20)
Threshold Level 3	(6, 10)	(12, 20)
Threshold Level 4	(8, 10)	(16, 20)

Table 3.2. *Simulation parameters for varying the PCA node density in a MANET size 50 in a 1500 X 300 network field.*

3.12 Comparison with Existing Methods

As shown in our results in Figures 3.5 to 3.36, we observed differing certification success ratios by varying threshold levels. Our contribution to the existing solutions [5, 7, 9, 12]

Fixed Simulation Parameters	
Mobile Node Speeds (m/s)	10, 20, 30, 40,50 and 60
Pause Times (sec.)	0, 30, 60, 90, 120, 300, 600 and 900
Total Simulation Time (sec.)	900
Packet Sending Rate (packets per second)	4
Packet Size (KB)	512
Medium Access Control (MAC)	802.11
Propagation Model	Two Ray Ground
Antenna	Omni-directional Antenna

Table 3.3. *Overall fixed simulation parameters for all MANET setups.*

is to categorize security needs according to application. By decresing the threshold level, the availability of security services will increase. Our improvement can be analyzed from two different perspectives: dividing security provided into several levels and increasing availability.

- **Dividing Security into Several Levels:** In previous work [5, 7, 9, 12], Zhou and Haas and S. Yi and R. Kravets designed their key management framework on one threshold level. By increasing the number of levels, we provide a hierarchical security approach from the application perspective. It is noteworthy that there is a trade-off between threshold levels m_t, the security provided by our key management technique, and availability of the security services. The specific number of PCA nodes or threshold levels represent the security level of our key management technique. Lowering the minimum required PCA nodes means lowering the security provided by our key management technique. An attacker needs to compromise the minimum specific number of PCA nodes to gain the control of MANET. If we increase the minimum required PCA nodes in the network, it will increase the security level, since the attacker needs to break into more nodes to gain control of security services. At the same time by increasing the security level, the security services becomes less available. We demonstrated this security - availability trade-off in our results. For example, the certification success ratio for 10 mobile nodes with threshold level 1, (2, 6) moving with maximum speed 40 m/s in 300m x 300m field is approximately

99.5% as demonstrated in Figure 3.5. When we increase the threshold level from 1 to 2 (3, 6) in the same network set-up, the certification success ration decreases by approximately 1% to 98.5%. Our results in Figures 3.5 to 3.36 consistently show the same behavior. There is a minimal increase in complexity due to storing multiple keys.

- **Increasing Availability:** As seen in our results in Figures 3.5 to 3.36, the availability of security services provided by key management services is increased as a result of lowering threshold levels. Of course, there is a cost for decreasing threshold levels. The security provided by our key management technique will decrease as a result of lowering the threslhold levels(lowering the number of PCA nodes).

When mobile node speed increases, the certification success ratio drops approximately 1% for every 10 m/s increase in mobile node speed as illustrated in Figures 3.5 to 3.36. When mobile nodes have high mobility, meaning the pause time is close to 0 seconds, there are some initial fluctuations in the certification success ratio due to the high mobility and random events in the simulation environment. In our results, most of these fluctuations occured between 0 seconds and 120 seconds pause time. Comparatively, when the nodes maintain a relatively constant position, there are very slight changes in the certification success ratio.

3.12.1 Effect of the Node Density and the Network Field Size

When we decrease n, the PCA node density in the network field for the MANET size 50, we lose 10 % of the certification success ratio as seen in Figures 3.29 to 3.36. This indicates that there is a trade-off between the node density and the certification success ratio. When we increase the network field size from 300m x 300m to 1500m x 300m as we increase MANET size from 10 to 50, we observe that for each incremental increase in MANET size we lose 1% of the certification success ratio, or 6% loss in total.

3.13 Summary

In Chapter 3, after explaining threshold cryptography, we surveyed the main existing key management solutions for MANETs. We described our proposed solution, which extends [5, 7, 9, 12] and explained how our approach categorizes the security needs. The hierarchical structure contributed by our solution enables a variety of applications to be categorized according to their security needs. We presented our experimental simulation results to show the effectiveness of our approach.

3.14 Simulation Results

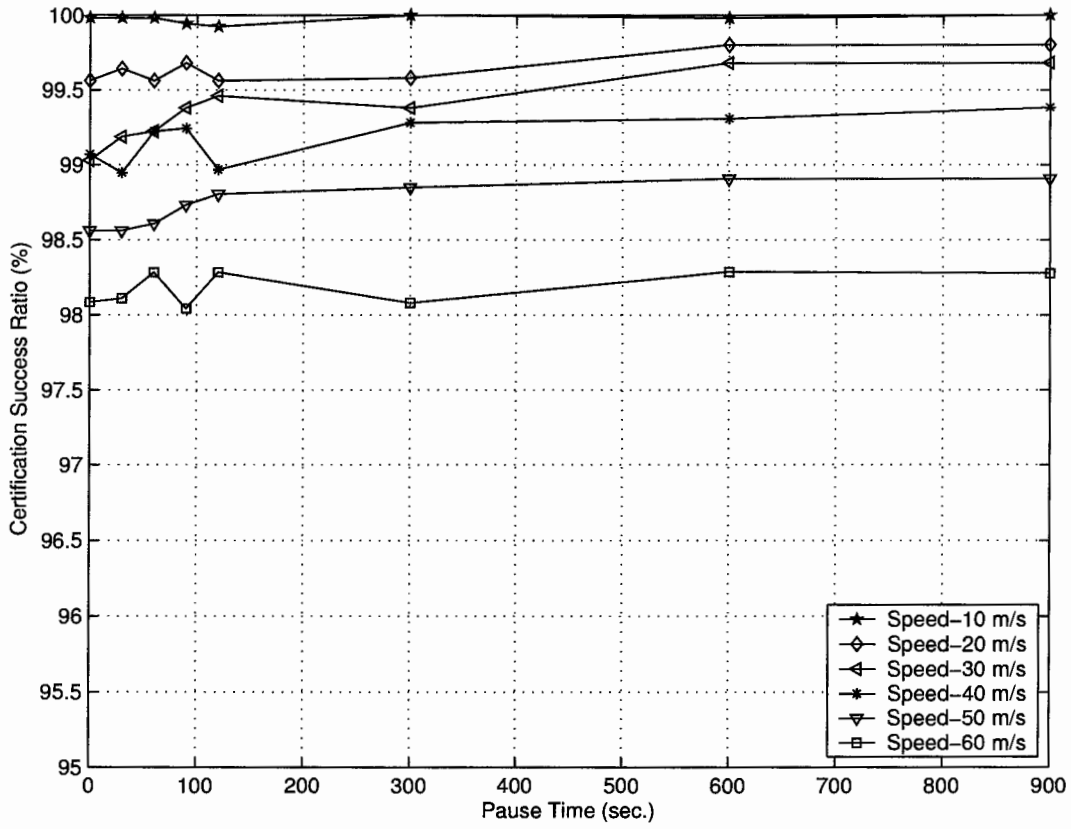

Figure 3.5. *Certification success ratio (%) for 10 mobile nodes with threshold level 1, (2, 6), in 300m X 300m field*

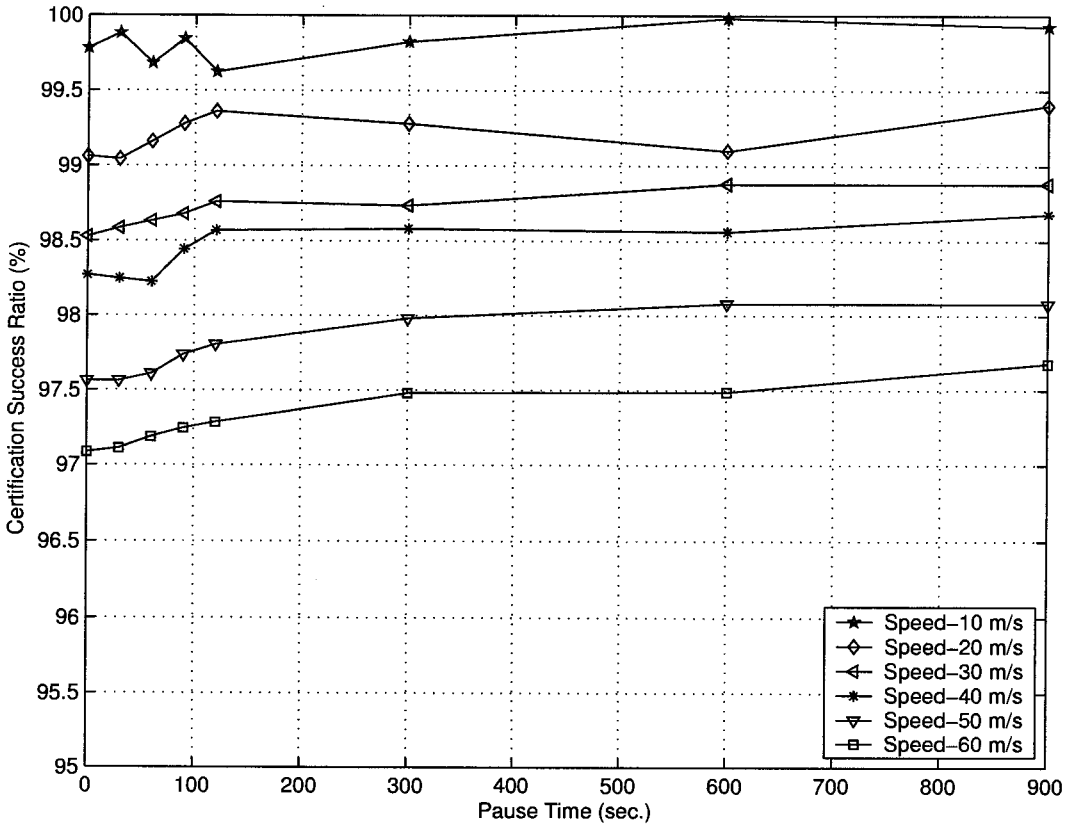

Figure 3.6. *Certification success ratio (%) for 10 mobile nodes with threshold level 2, (3, 6), in 300m X 300m field*

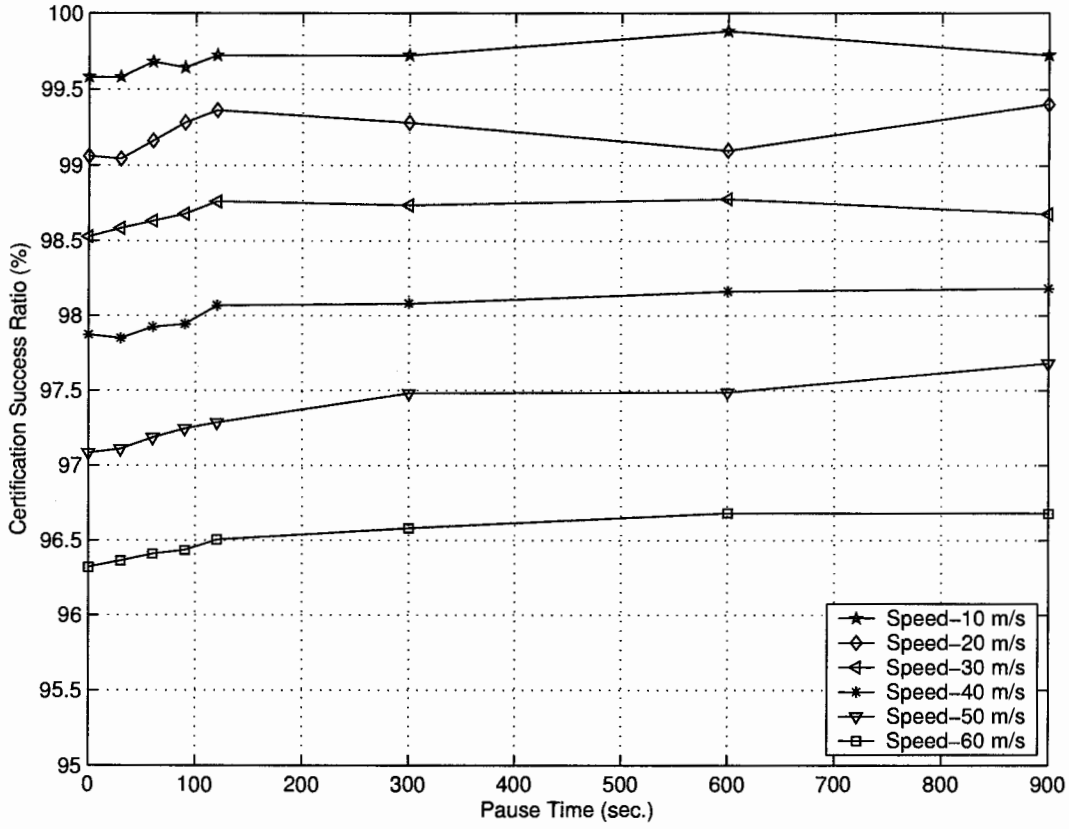

Figure 3.7. *Certification success ratio (%) for 10 mobile nodes with threshold level 3, (4, 6), in 300m X 300m field*

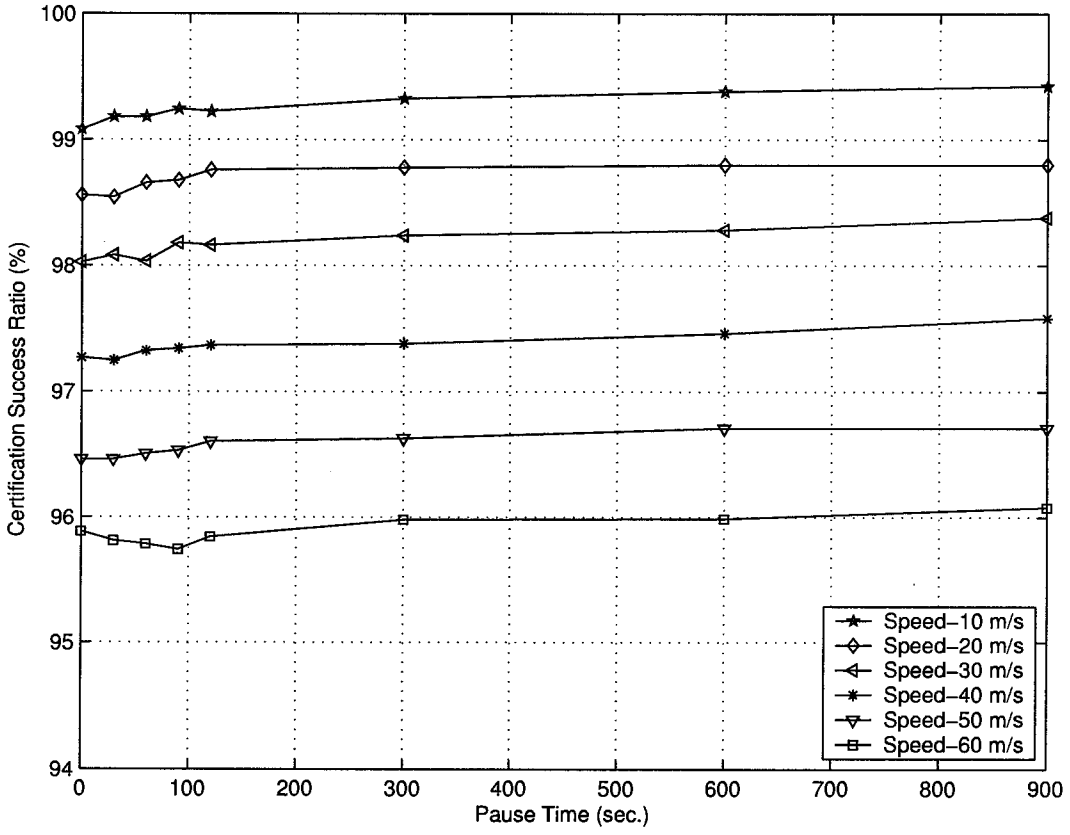

Figure 3.8. *Certification success ratio (%) for 10 mobile nodes with threshold level 4, (5, 6), in 300m X 300m field*

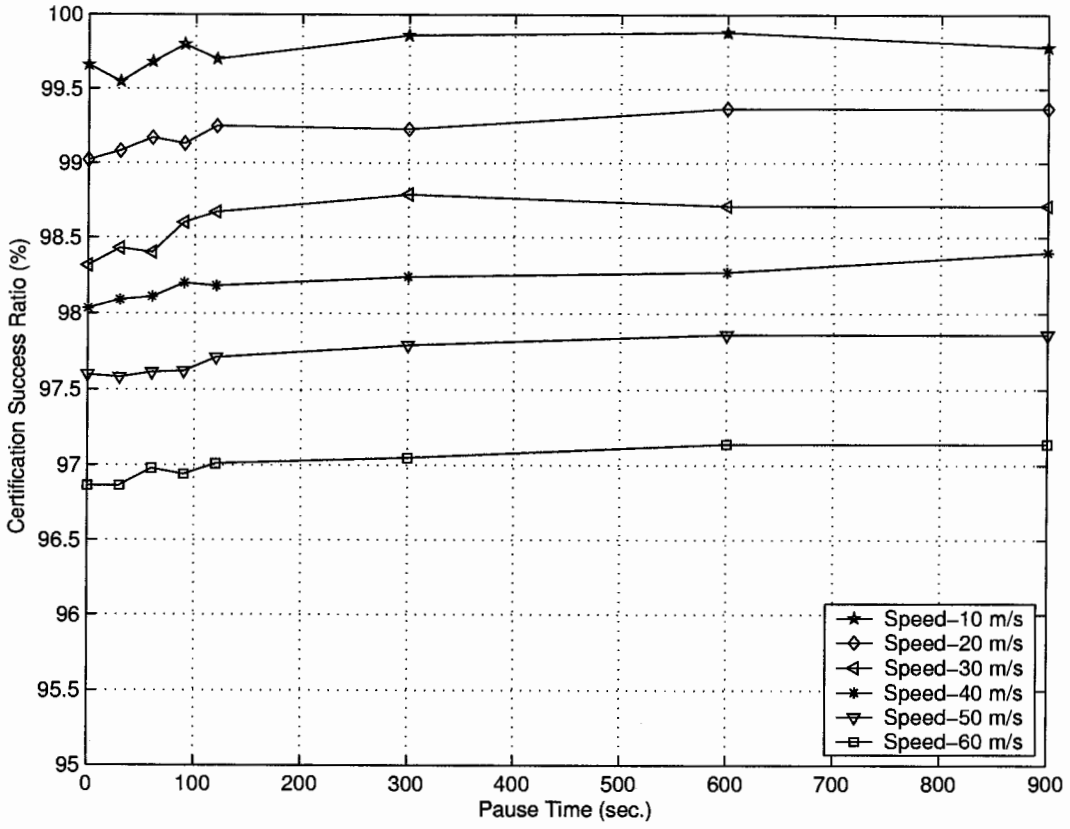

Figure 3.9. *Certification success ratio (%) for 20 mobile nodes with threshold level 1, (2, 12), in 600m X 300m field*

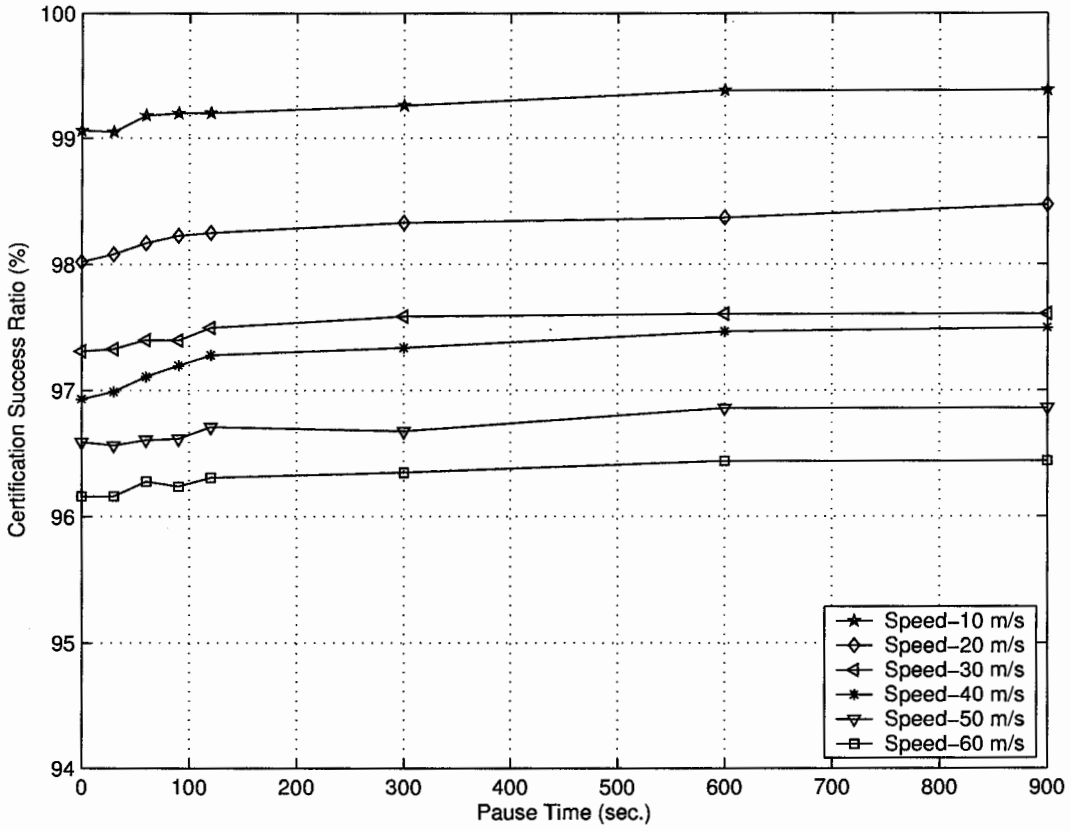

Figure 3.10. *Certification success ratio (%) for 20 mobile nodes with threshold level 2, (4, 12), in 600m X 300m field*

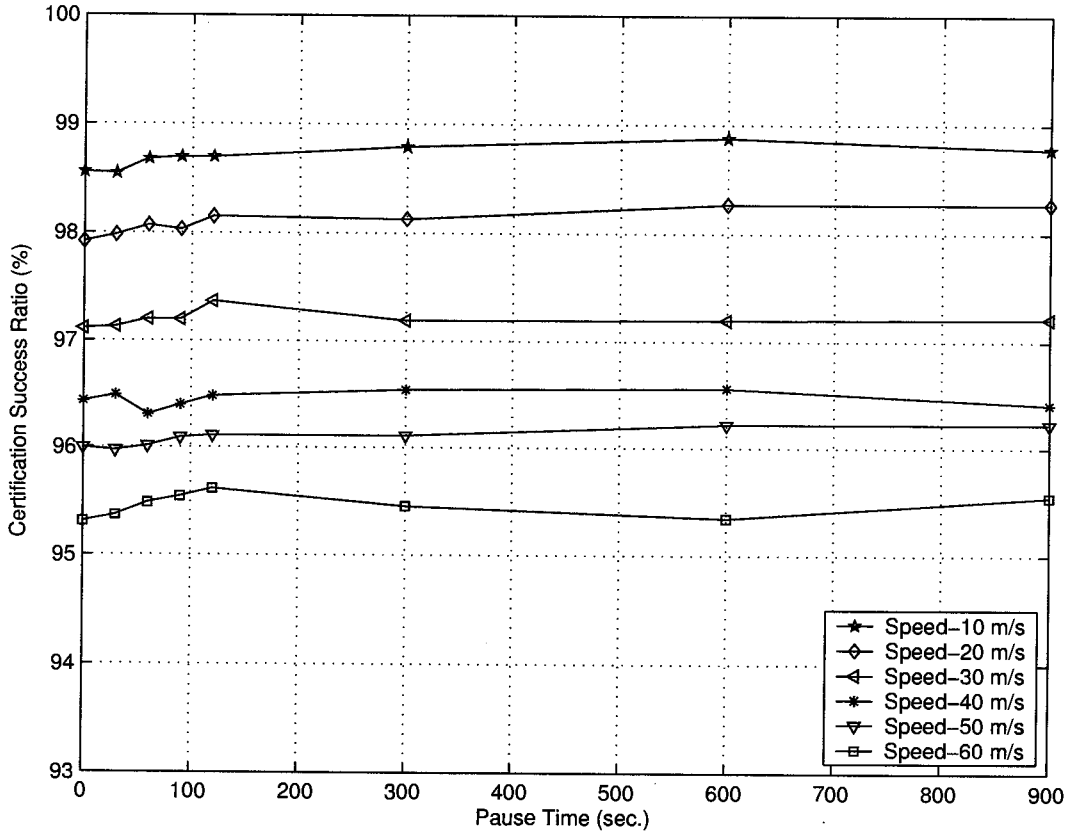

Figure 3.11. *Certification success ratio (%) for 20 mobile nodes with threshold level 3, (6, 12), in 600m X 300m field*

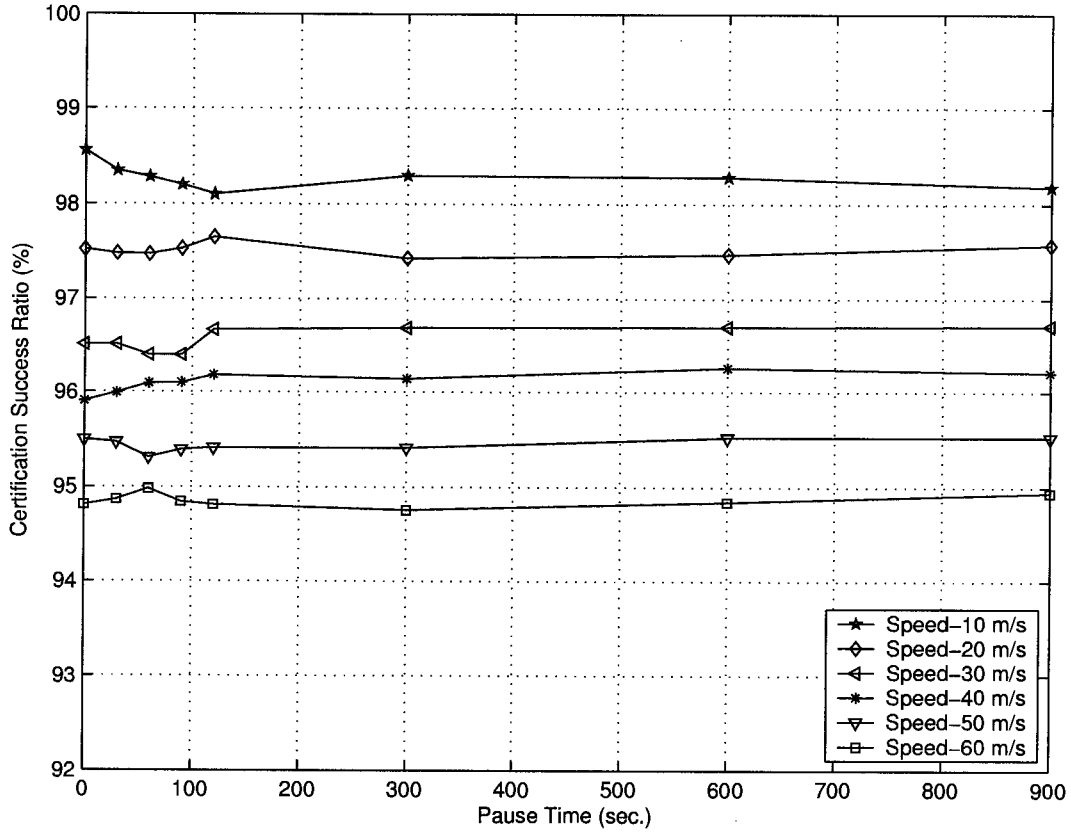

Figure 3.12. *Certification success ratio (%) for 20 mobile nodes with threshold level 4, (8, 12), in 600m X 300m field*

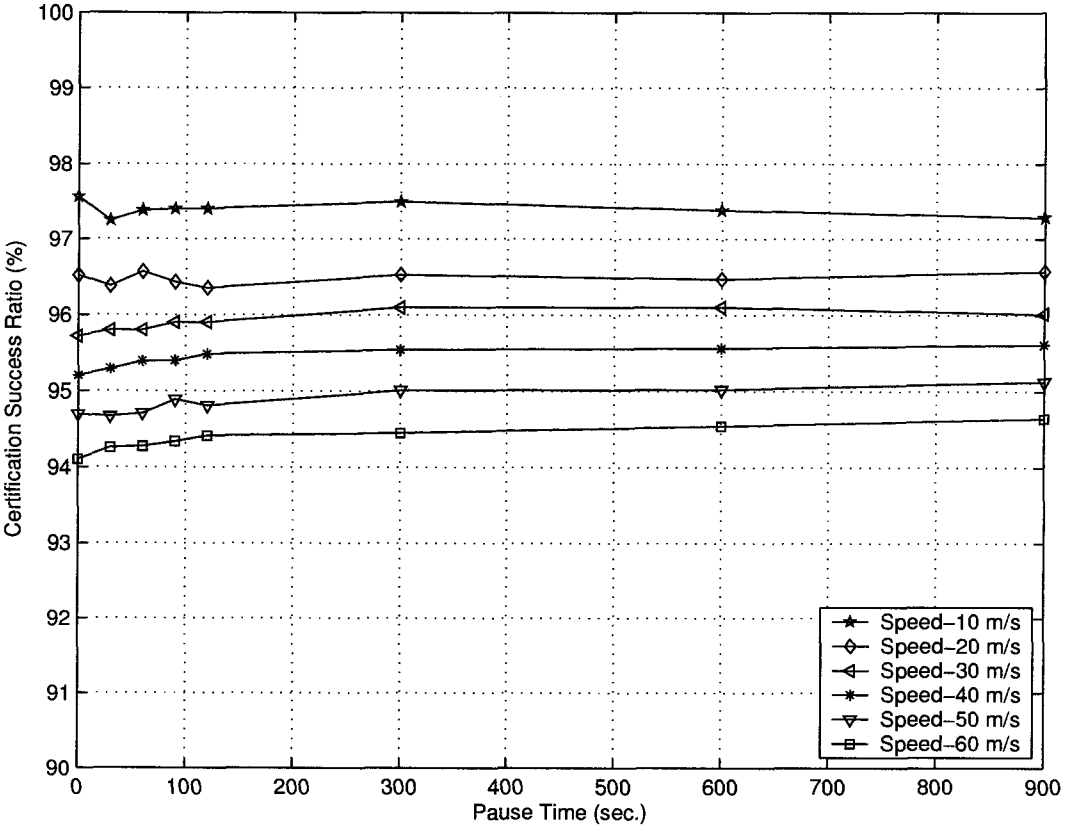

Figure 3.13. *Certification success ratio (%) for 20 mobile nodes with threshold level 5, (10, 12), in 600m X 300m field*

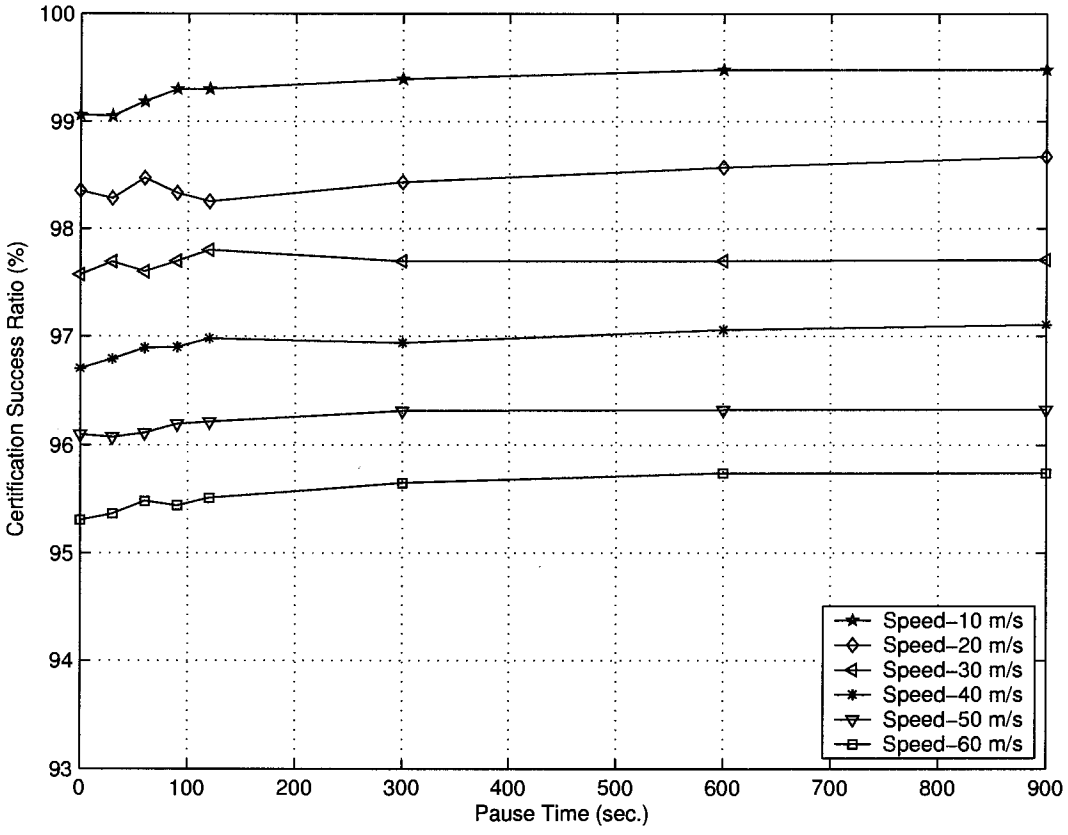

Figure 3.14. *Certification success ratio (%) for 30 mobile nodes with threshold level 1, (3, 18), in 900m X 300m field*

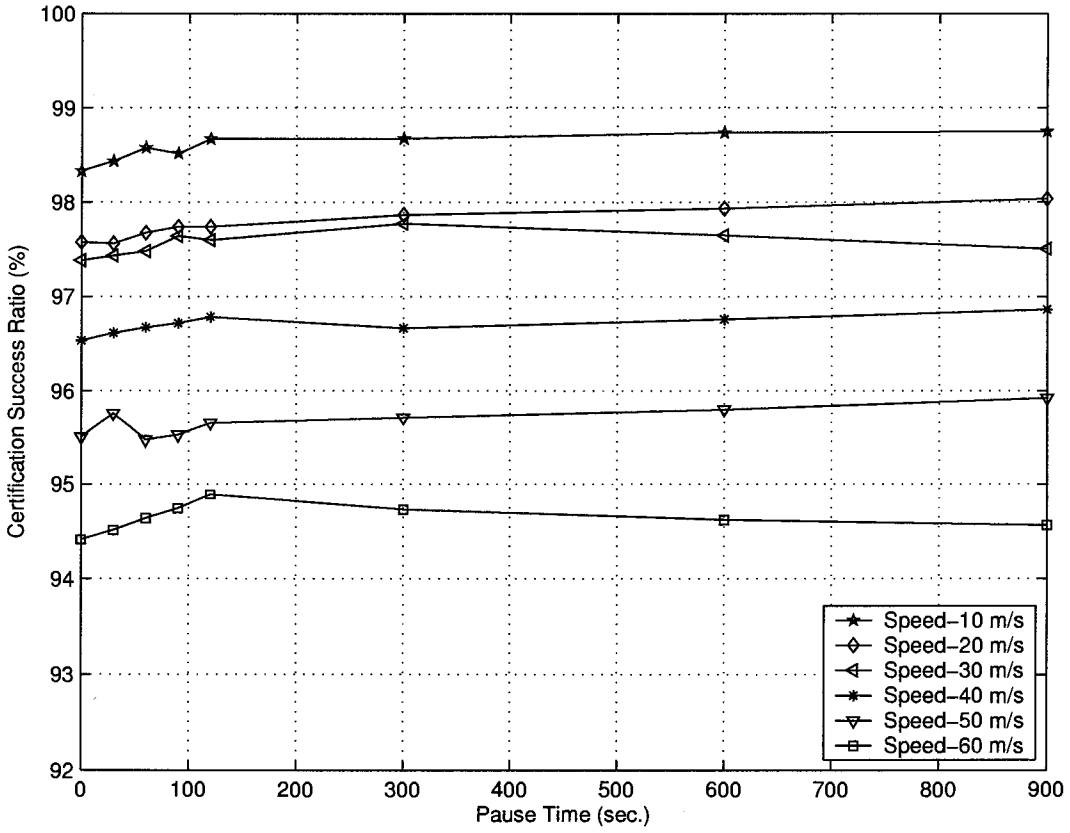

Figure 3.15. *Certification success ratio (%) for 30 mobile nodes with threshold level 2, (6, 18), in 900m X 300m field*

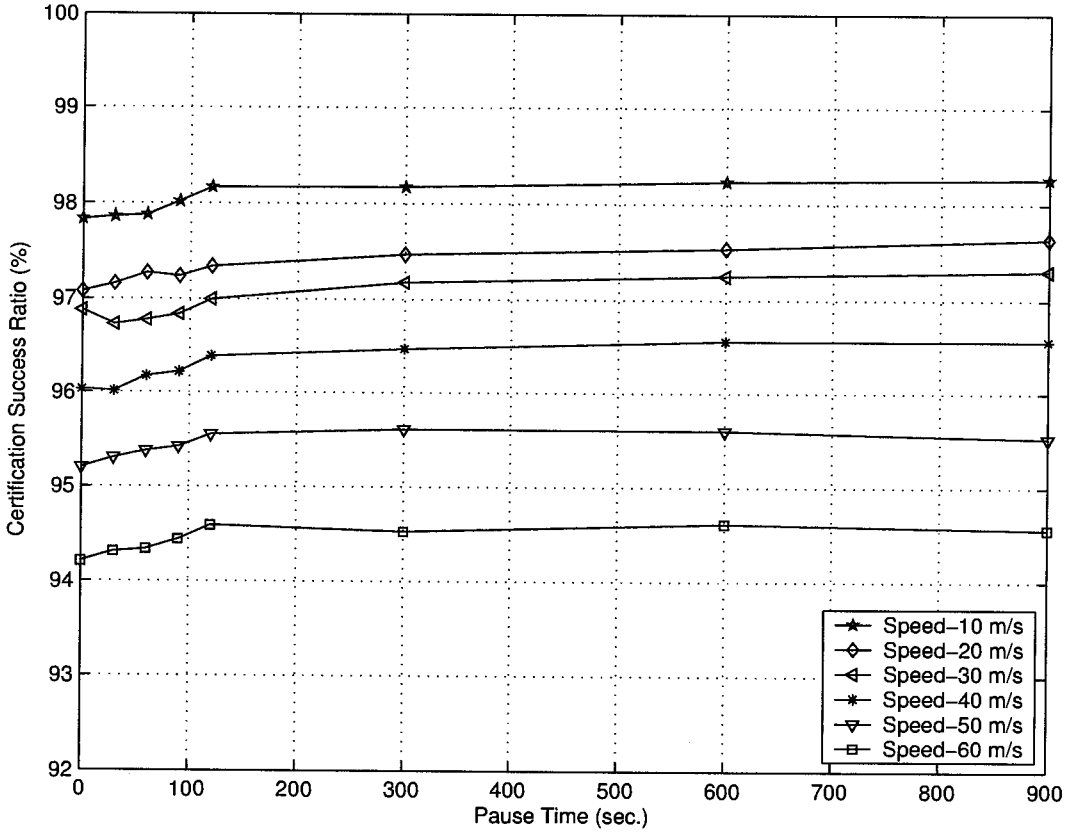

Figure 3.16. *Certification success ratio (%) for 30 mobile nodes with threshold level 3, (9, 18), in 900m X 300m field*

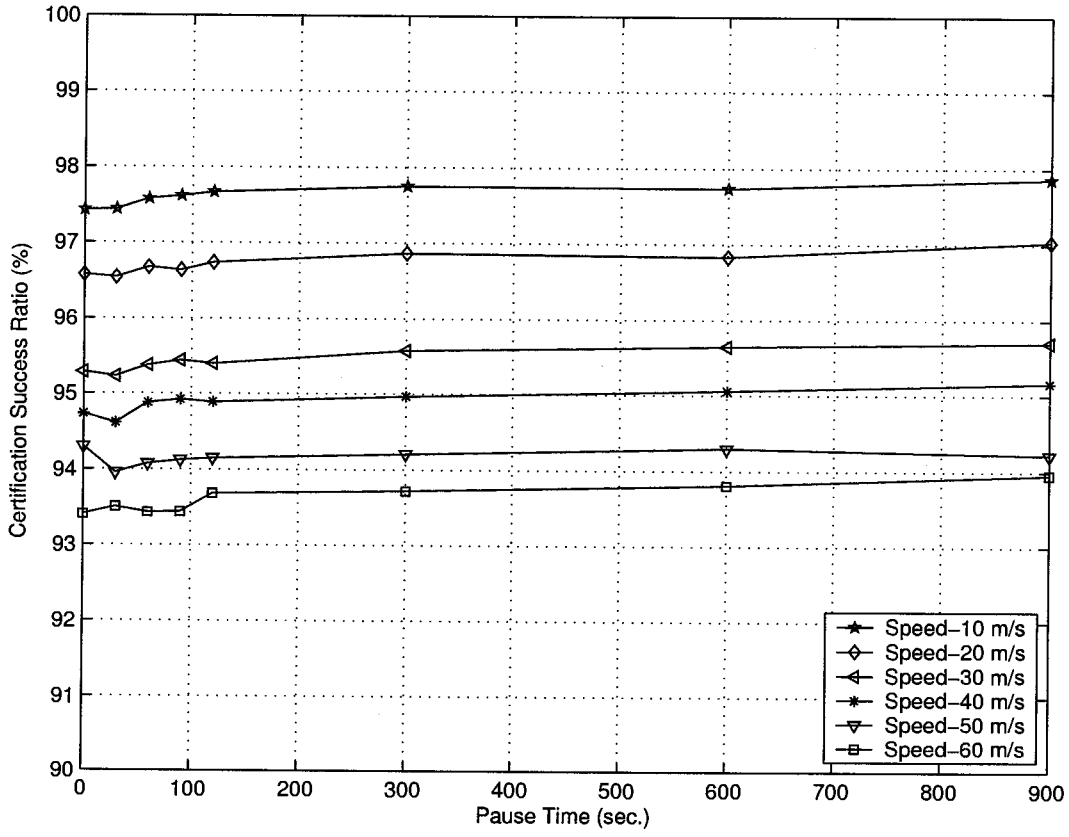

Figure 3.17. *Certification success ratio (%) for 30 mobile nodes with threshold level 4, (12, 18), in 900m X 300m field*

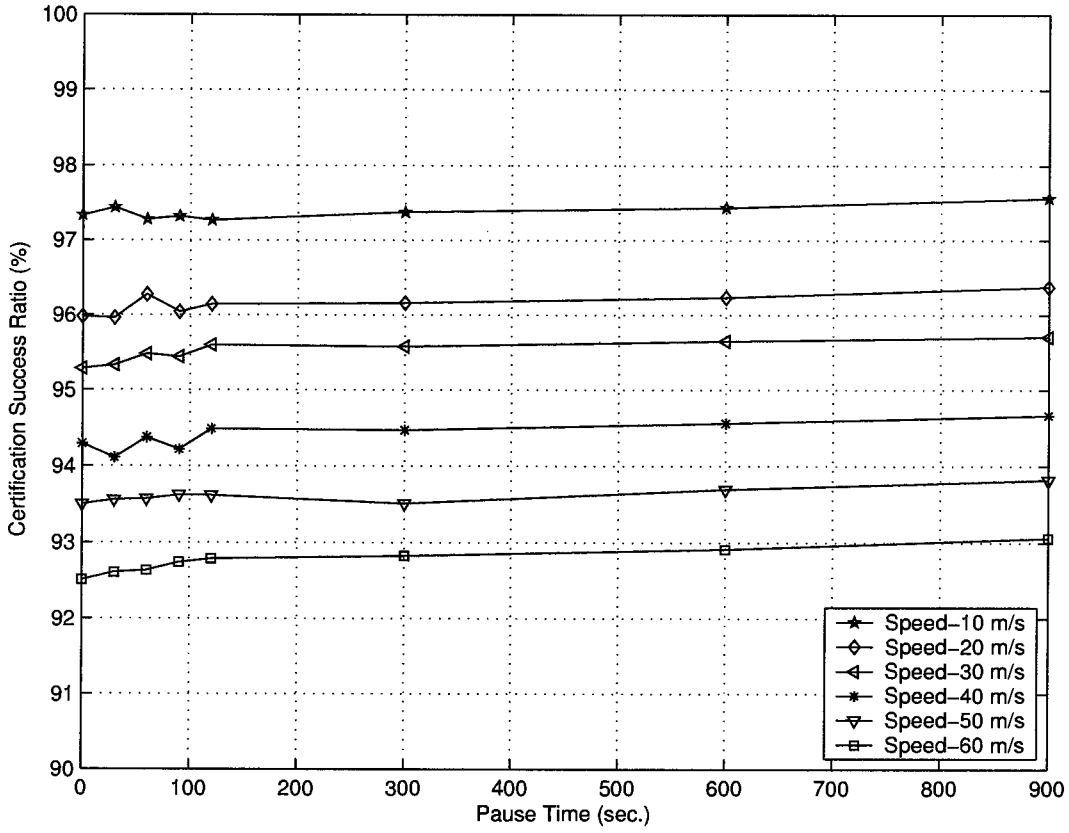

Figure 3.18. *Certification success ratio (%) for 30 mobile nodes with threshold level 5, (15, 18), in 900m X 300m field*

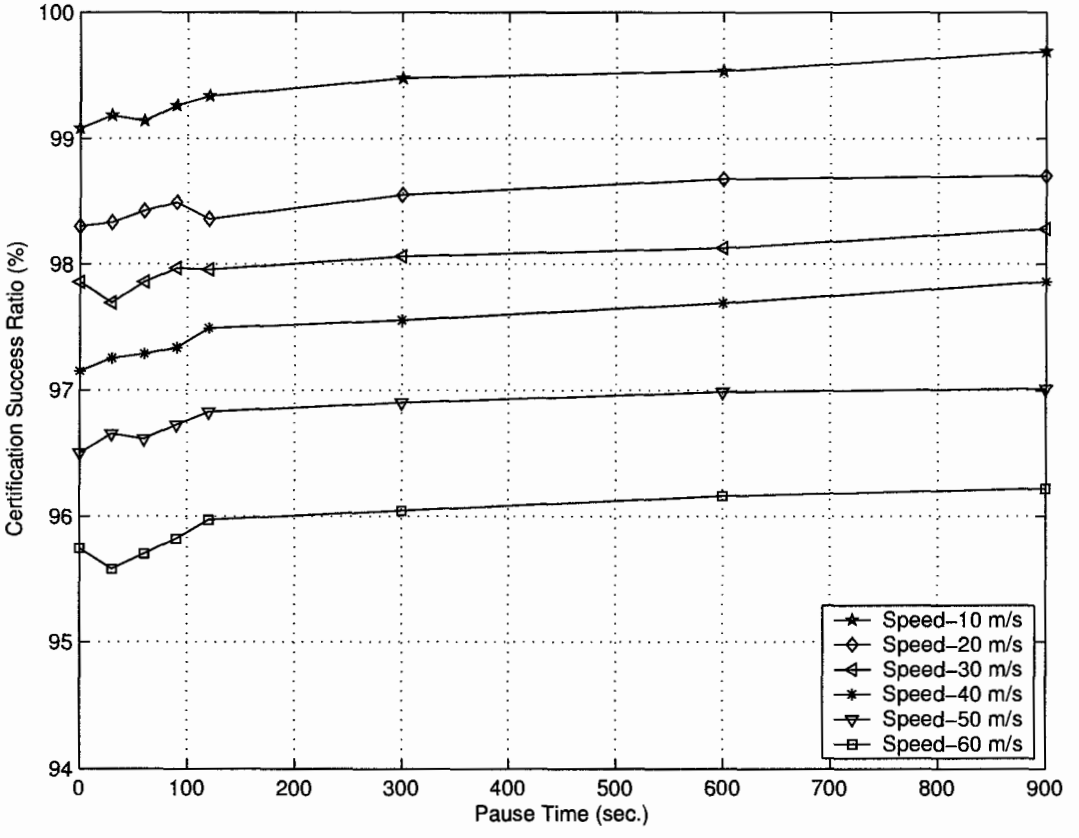

Figure 3.19. *Certification success ratio (%) for 40 mobile nodes with threshold level 1, (4, 24), in 1200m X 300m field*

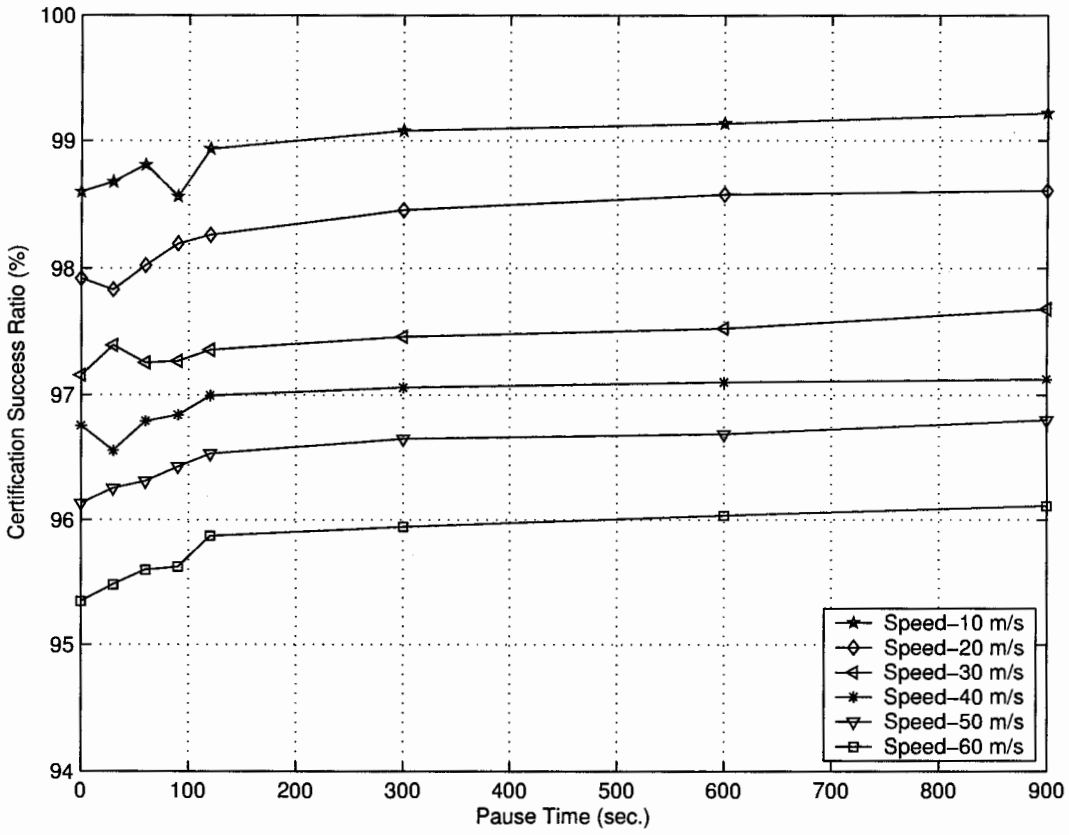

Figure 3.20. *Certification success ratio (%) for 40 mobile nodes with threshold level 2, (8, 24), in 1200m X 300m field*

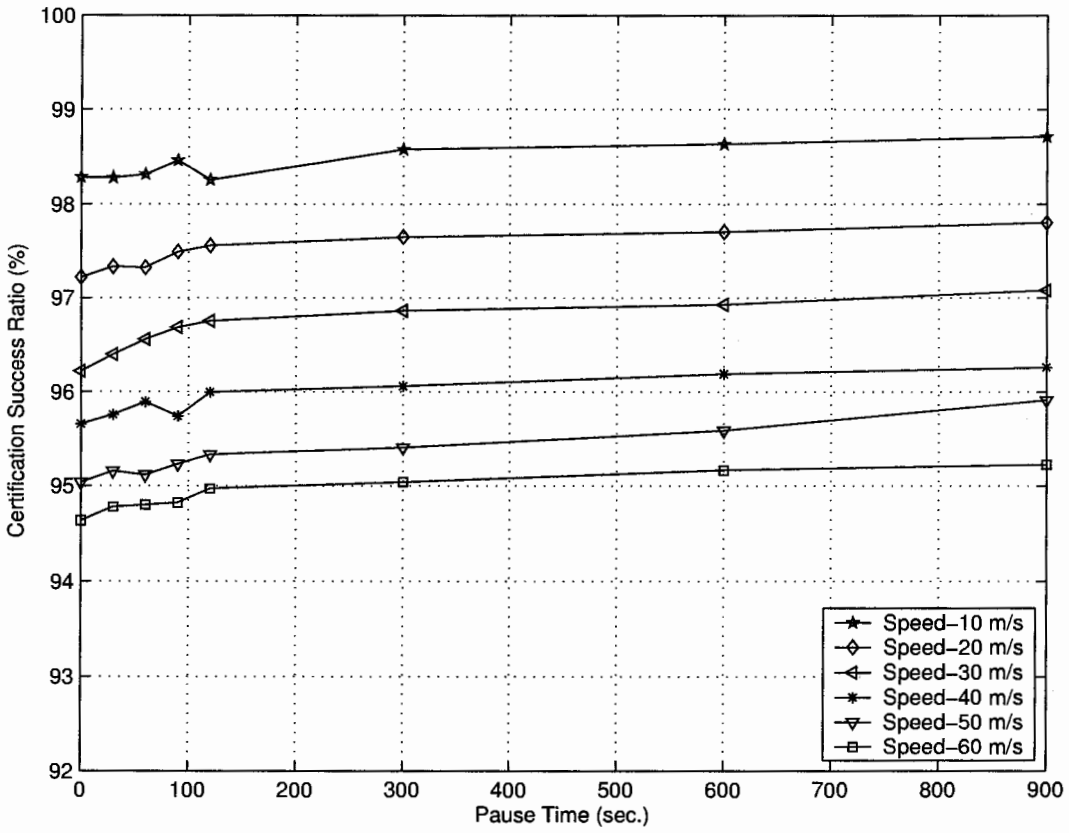

Figure 3.21. *Certification success ratio (%) for 40 mobile nodes with threshold level 3, (12, 24), in 1200m X 300m field*

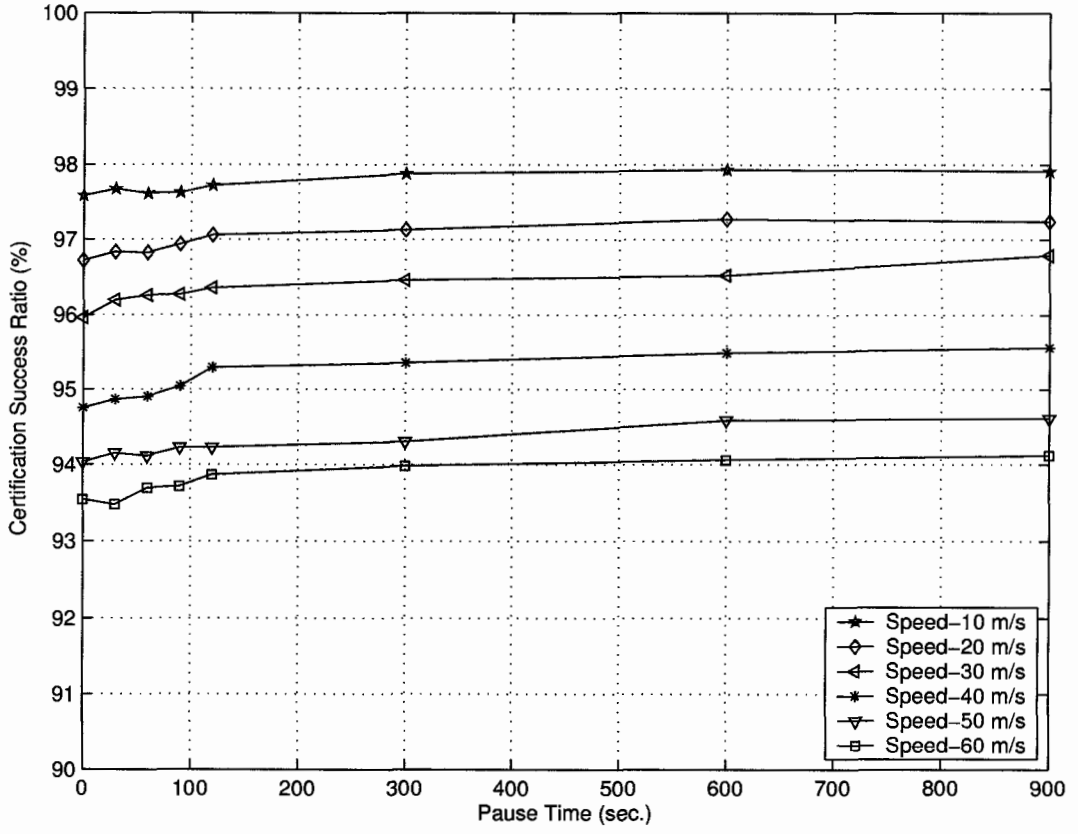

Figure 3.22. *Certification success ratio (%) for 40 mobile nodes with threshold level 4, (16, 24), in 1200m X 300m field*

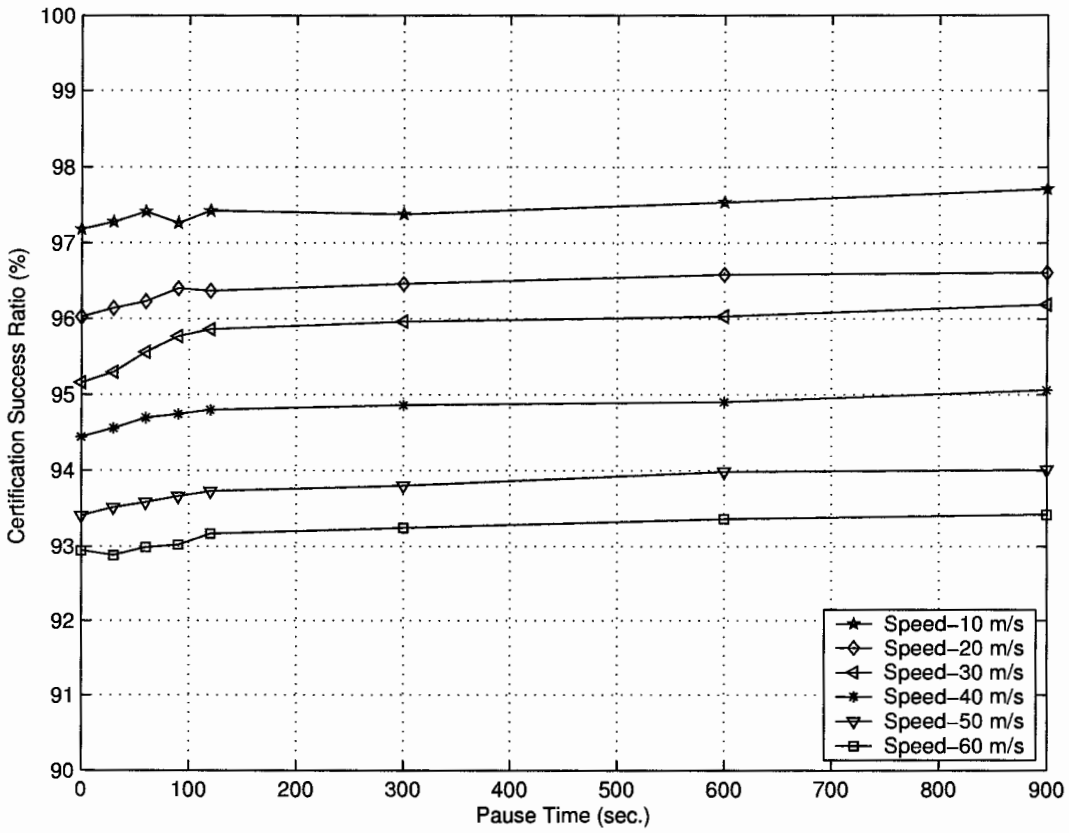

Figure 3.23. *Certification success ratio (%) for 40 mobile nodes with threshold level 5, (20, 24), in 1200m X 300m field*

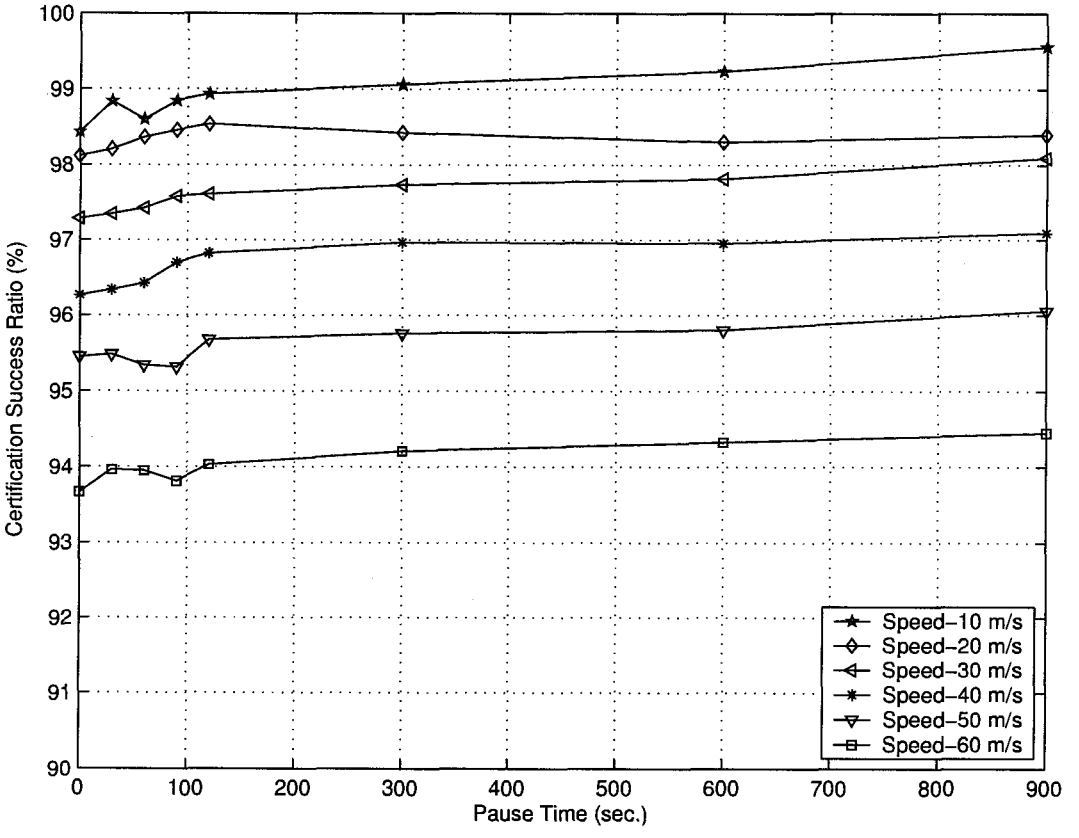

Figure 3.24. *Certification success ratio (%) for 50 mobile nodes with threshold level 1, (5, 30), in 1500m X 300m field*

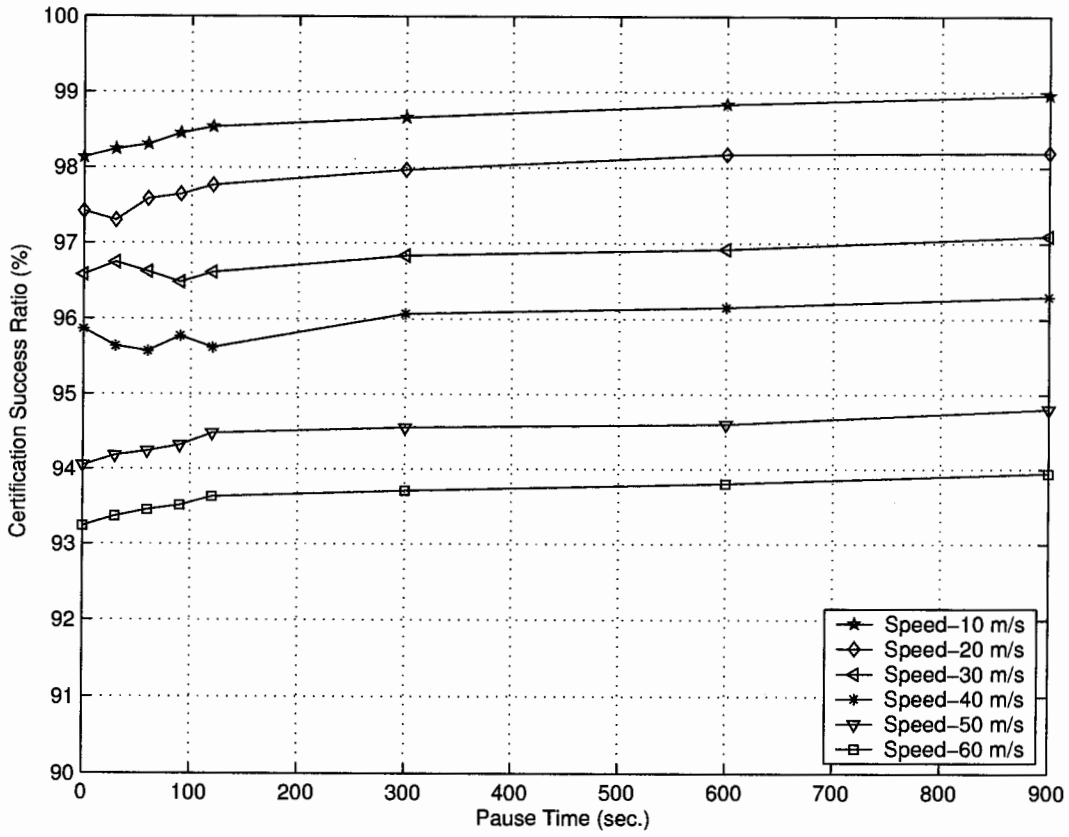

Figure 3.25. *Certification success ratio (%) for 50 mobile nodes with threshold level 2, (10, 30), in 1500m X 300m field*

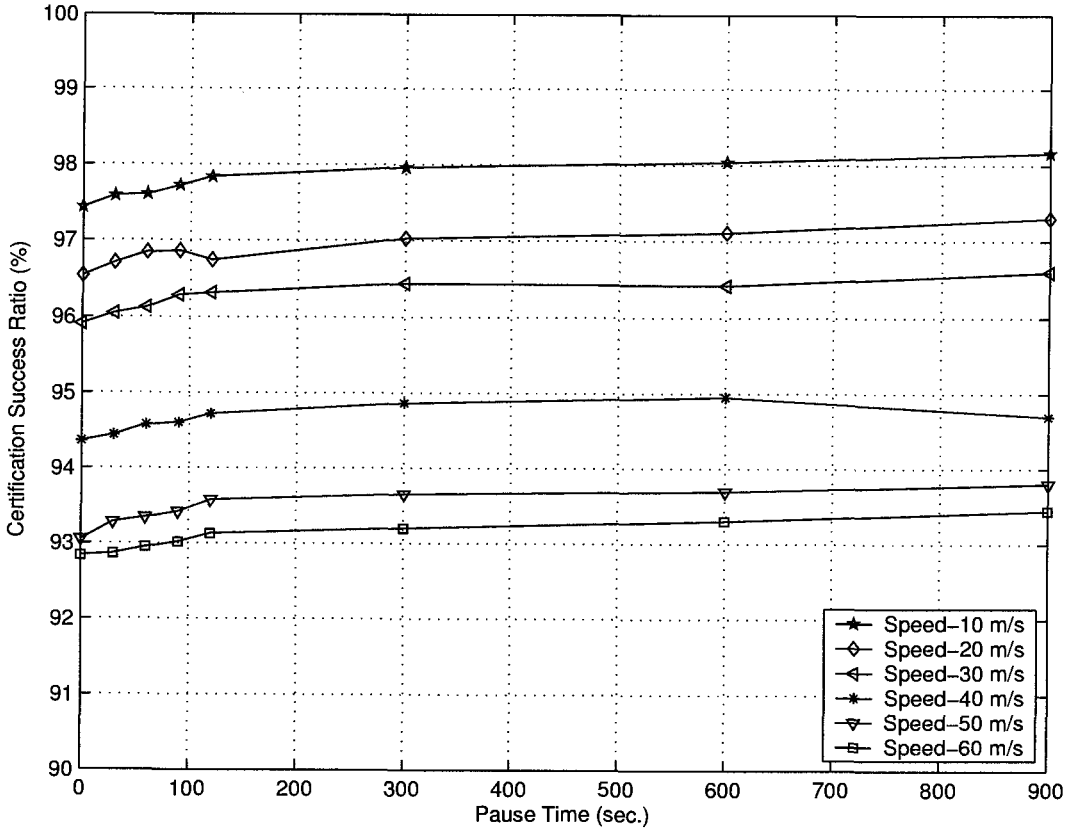

Figure 3.26. *Certification success ratio (%) for 50 mobile nodes with threshold level 3, (15, 30), in 1500m X 300m field*

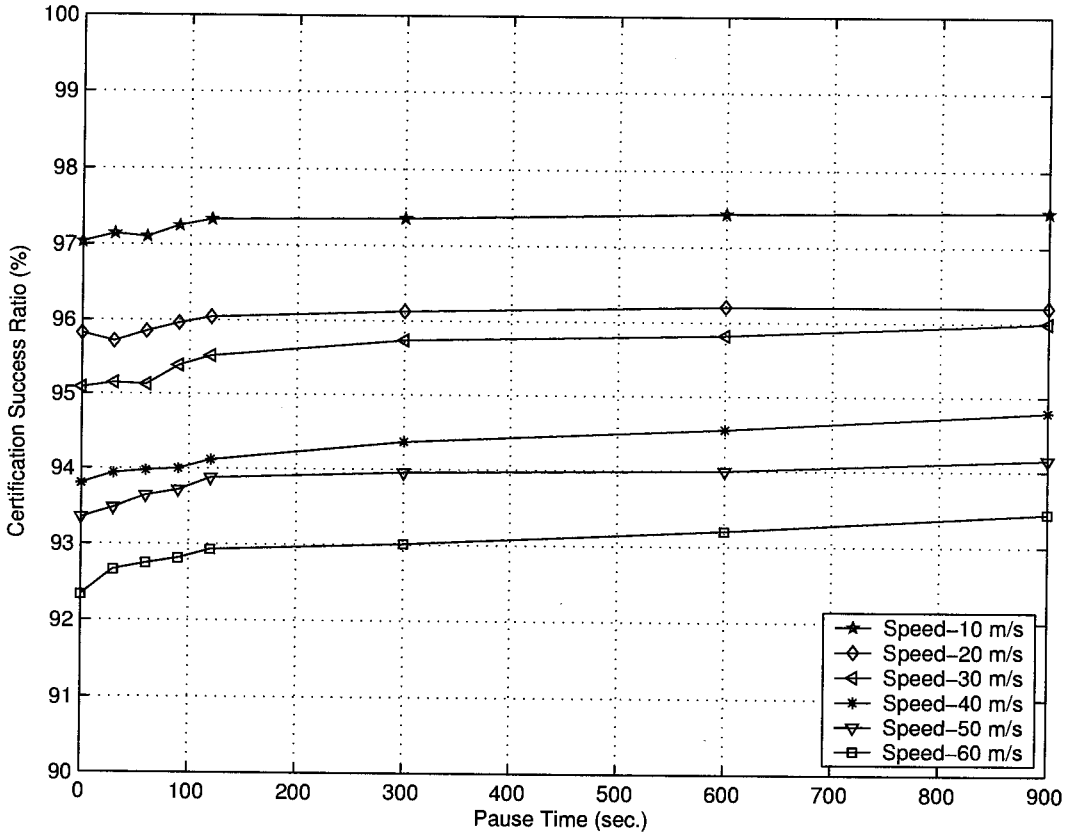

Figure 3.27. *Certification success ratio (%) for 50 mobile nodes with threshold level 4, (20, 30), in 1500m X 300m field*

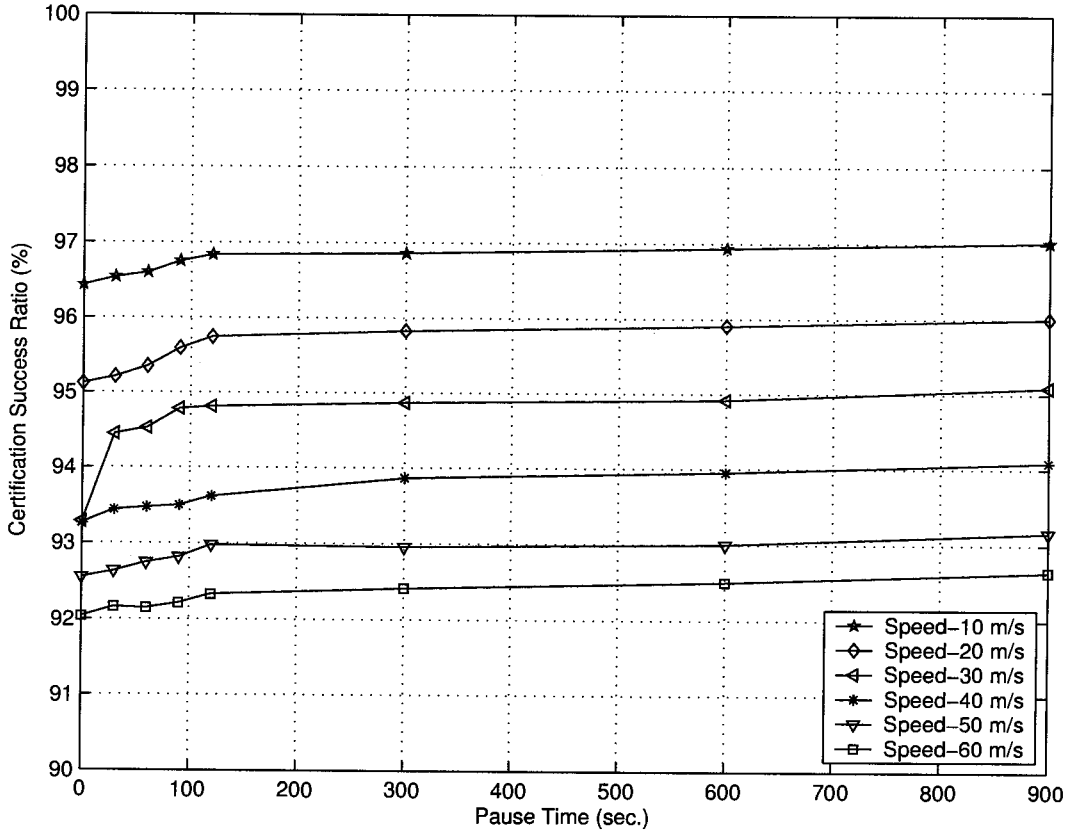

Figure 3.28. *Certification success ratio (%) for 50 mobile nodes with threshold level 5, (25, 30), in 1500m X 300m field*

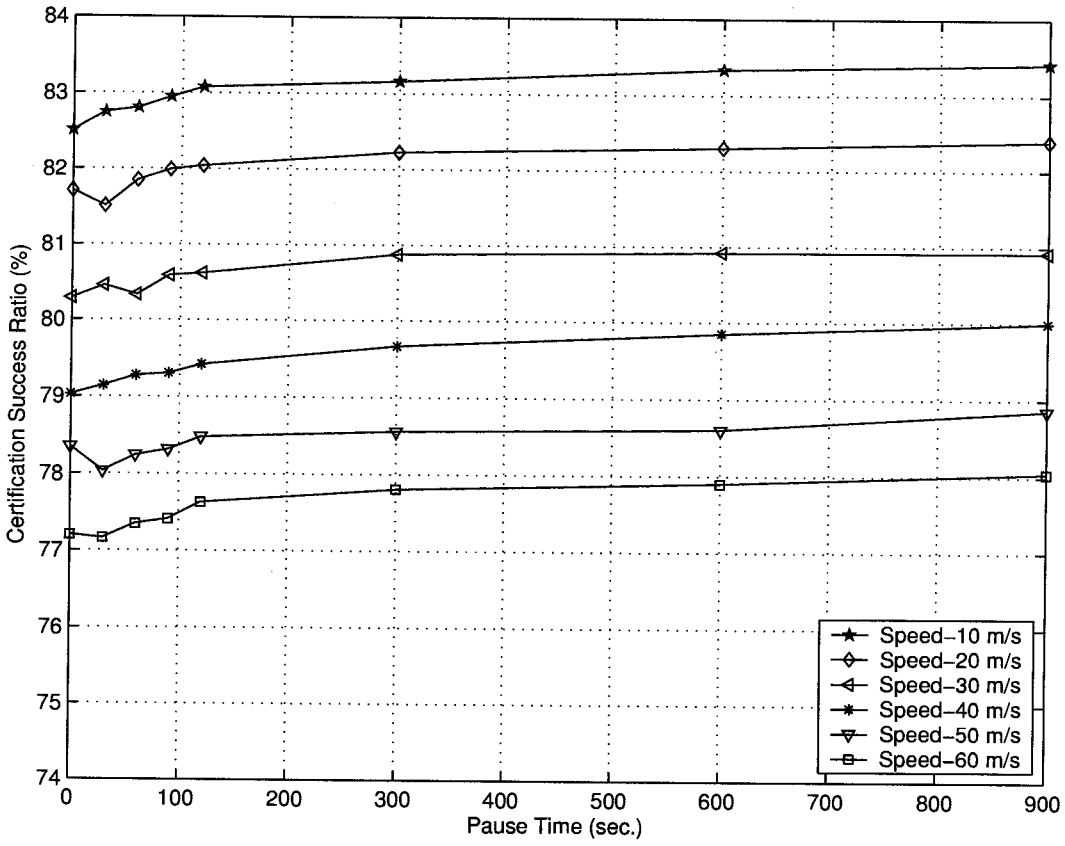

Figure 3.29. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (4, 20), in 1500m X 300m field*

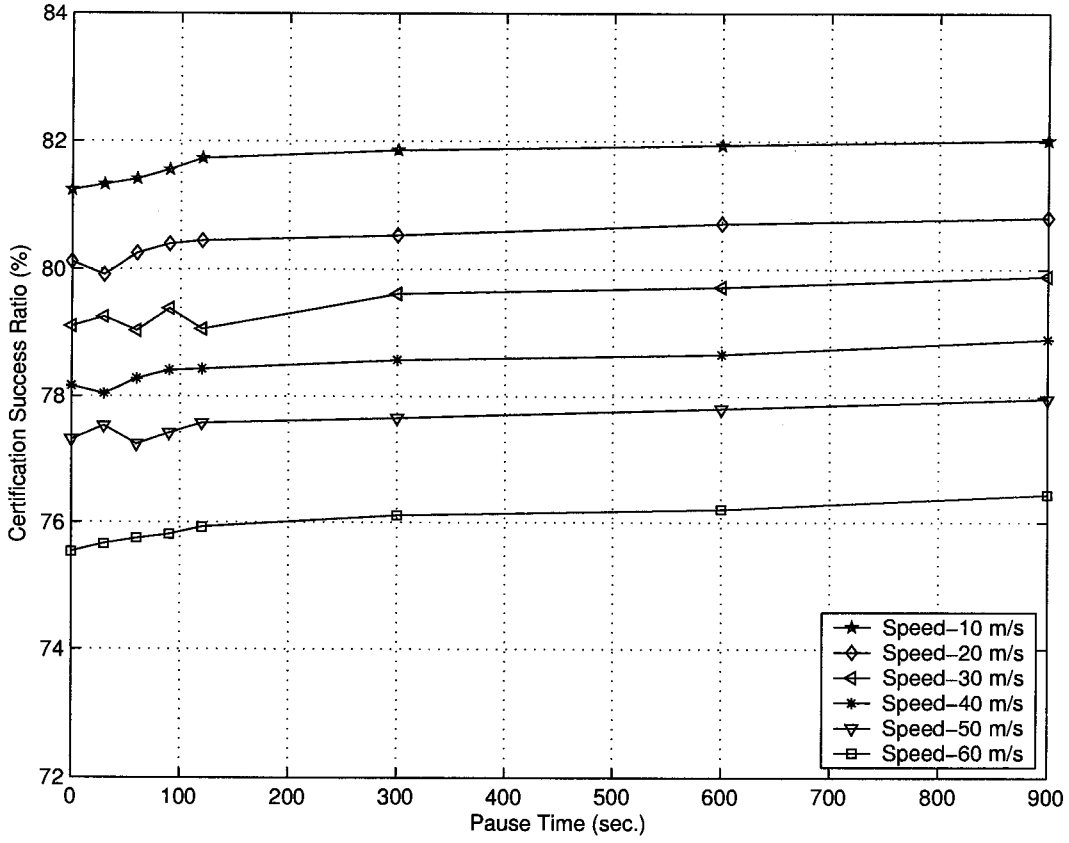

Figure 3.30. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (8, 20), in 1500m X 300m field*

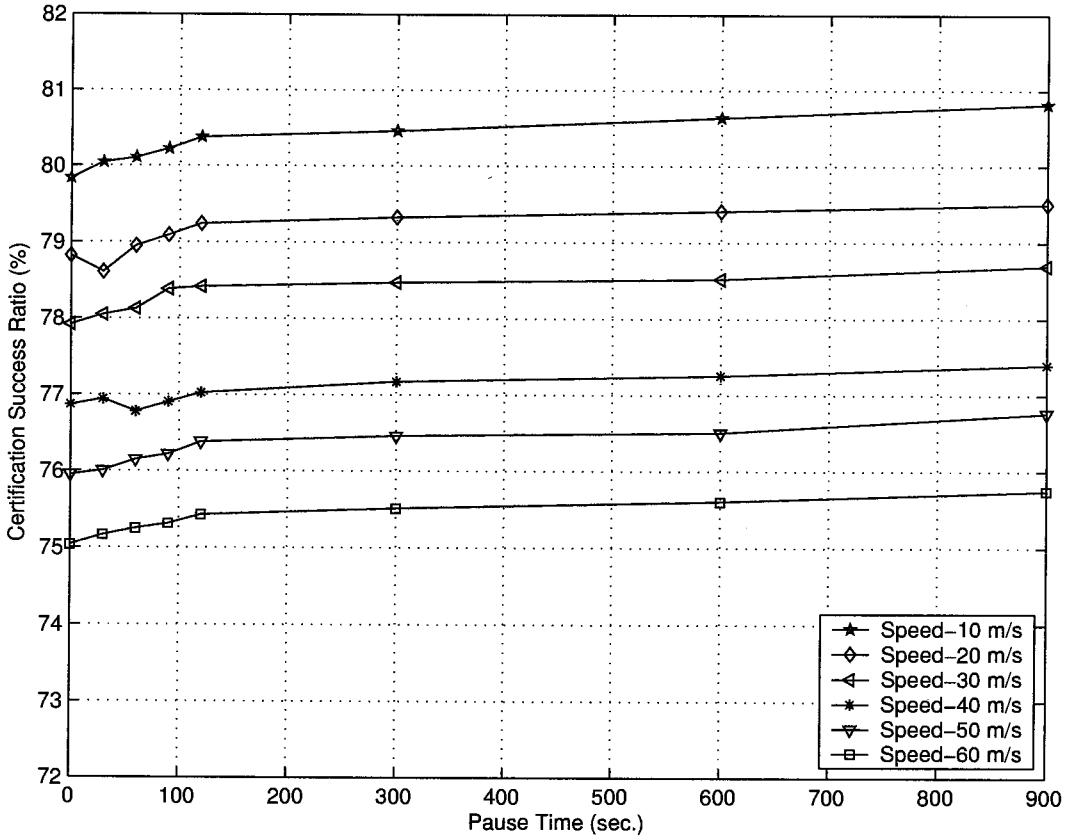

Figure 3.31. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (12, 20), in 1500m X 300m field*

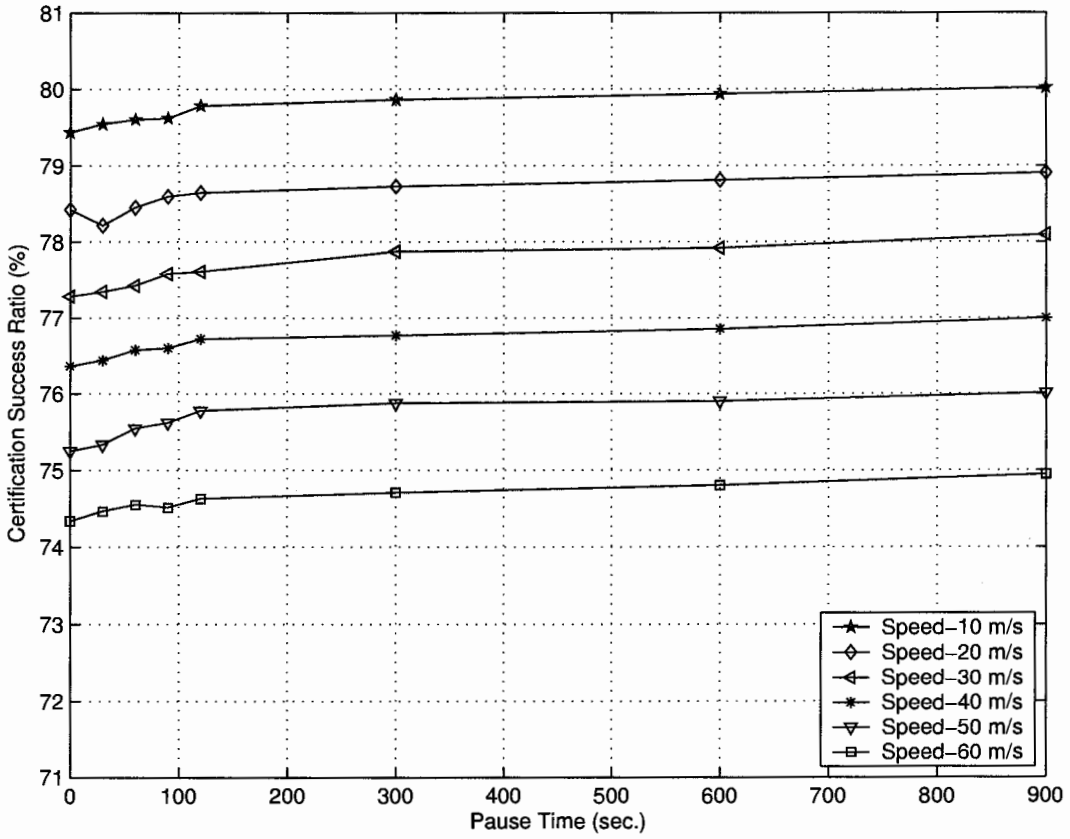

Figure 3.32. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (16, 20), in 1500m X 300m field*

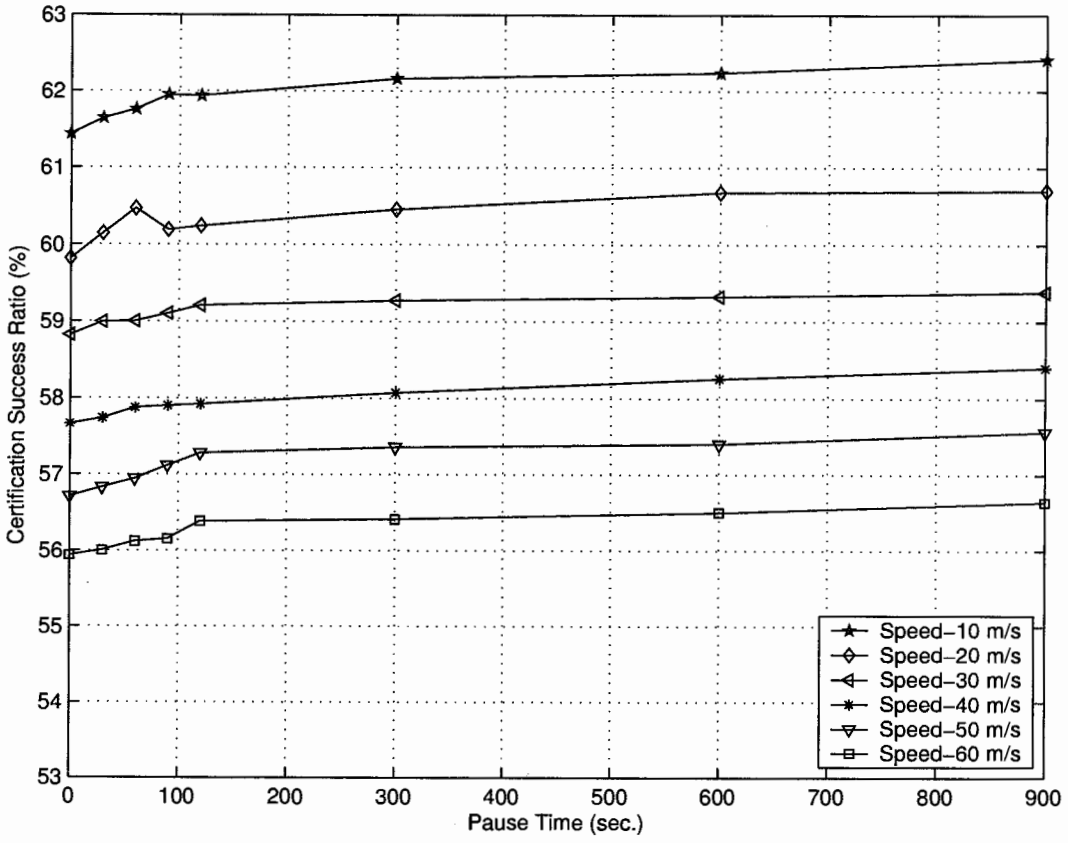

Figure 3.33. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (2, 10), in 1500m X 300m field*

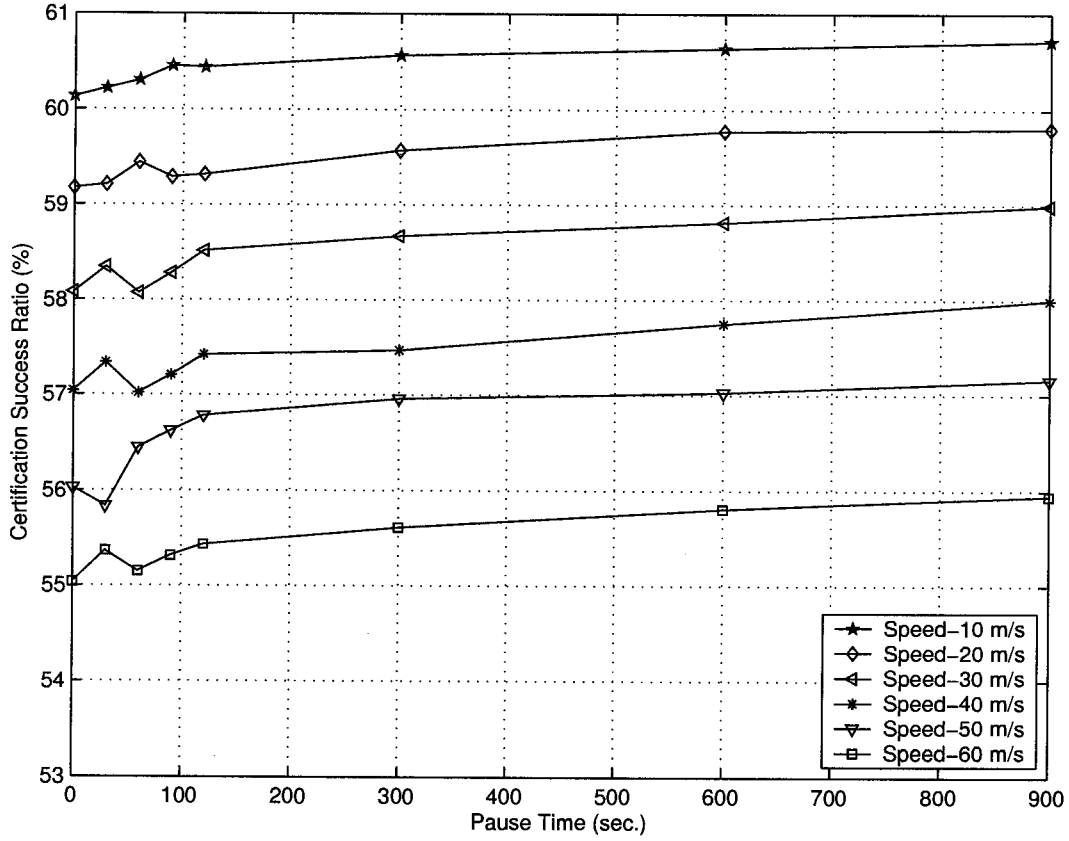

Figure 3.34. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (4, 10), in 1500m X 300m field*

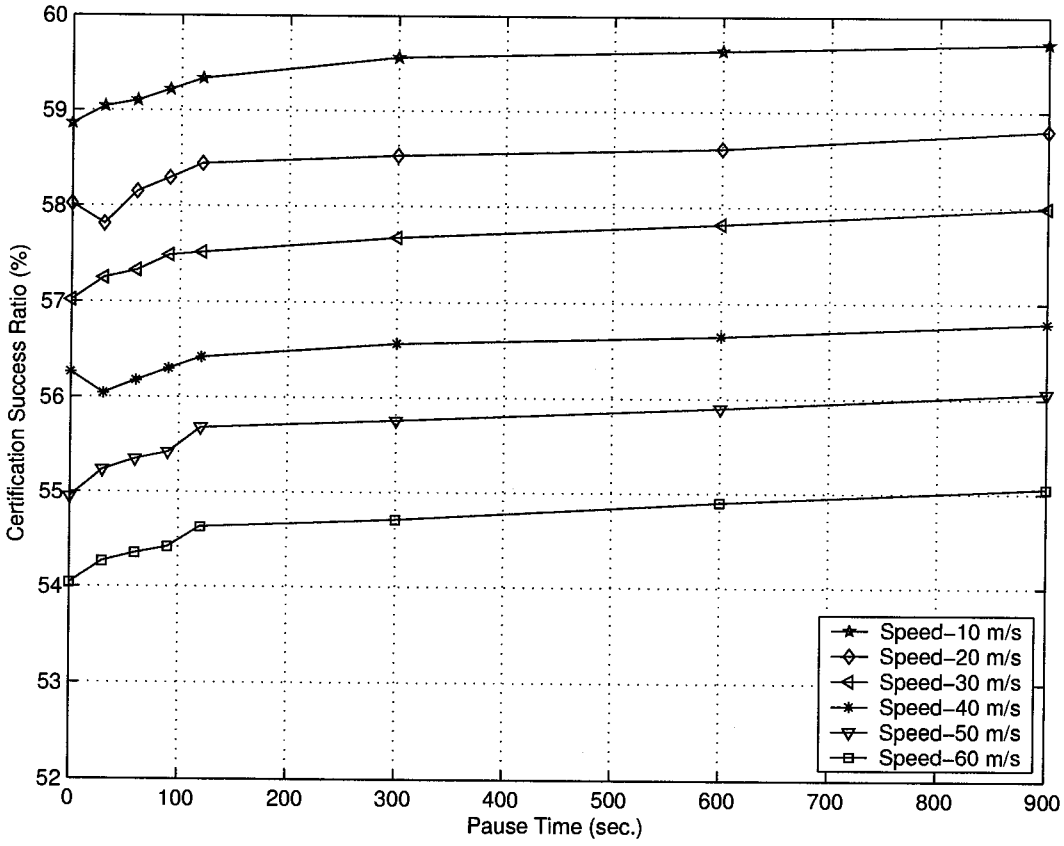

Figure 3.35. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (6, 10), in 1500m X 300m field*

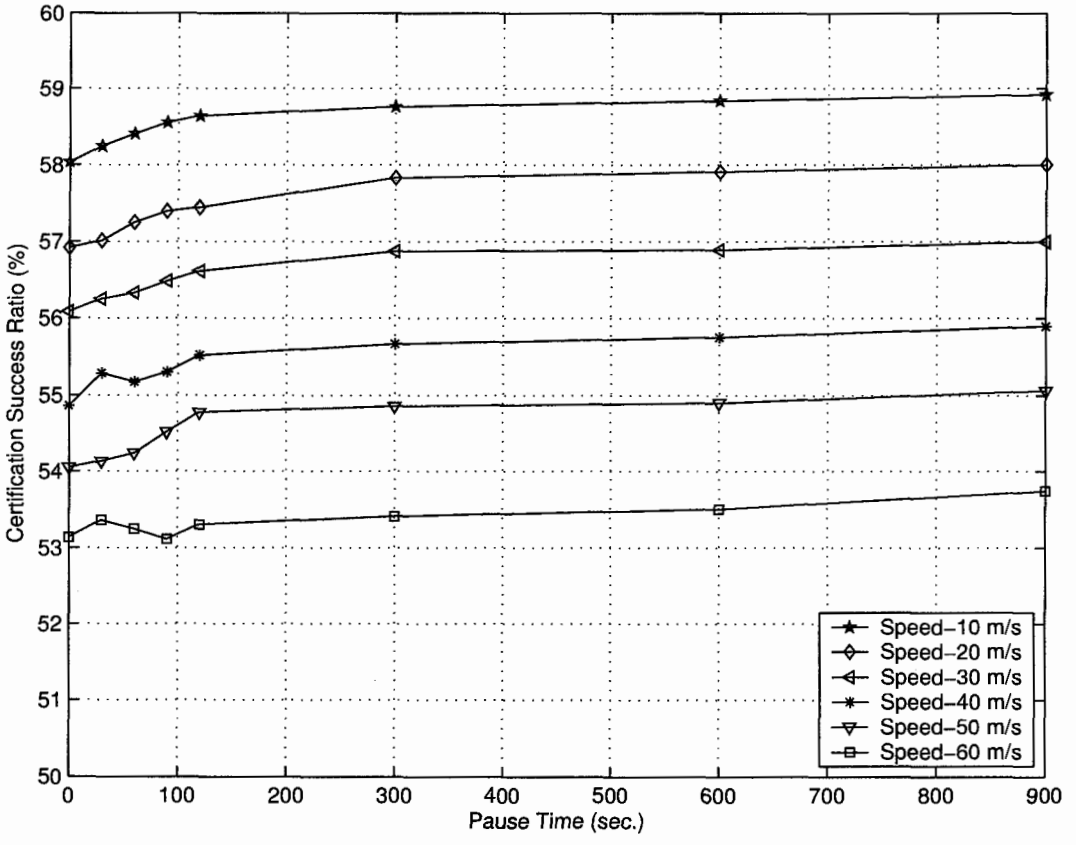

Figure 3.36. *Certification success ratio (%) for 50 mobile nodes, varying PCA node density, (8, 10), in 1500m X 300m field*

Chapter 4

Conclusion

Mobile Ad-hoc Networks are a new networking technology. We have shown how they differ from traditional networks and how traditional key management solutions often are unable to provide the same security in MANETs. We have pointed out major threats to security in MANETs and crucial objectives to securing these networks. We surveyed the major key management solutions in the literature and presented our proposed solution. There are many promising security services and applications that can be deployed in MANETs with the support of PKI. We chose PKI to solve the security dilemma in MANETs because most of the routing protocols assume the existence of PKI support that can easily utilize our proposed solution. In this thesis, we presented a practical key management technique for MANETs by extending the solution proposed by Zhou and Haas [5].

In Chapter 1, we introduced MANETs and their applications. We discussed the main challenges presented by MANETs; namely, dynamic topology and the lack of a pre-existing central infrastructure to establish robust and reliable key management techniques to provide security services. In Chapter 2, after providing the necessary background information to understand the cryptographic terminologies used in this thesis, we clarified the needs for a TTP in the PKI and pointed out the importance of the problem of providing a suitable key management framework. In Chapter 3, after surveying in detail the major security solutions as proposed in the literature for MANETs, we extended the current proposed solution based on threshold cryptography, by adding a hierarchical structure to increase availability. Using

ns-2, we simulated the main structure of our extended key management technique.

Certification success ratio results from our simulations show the effectiveness of our proposed key management technique. We demonstrated that our key management scheme performs well with MANETs. In all simulations, we had at least a 92% certification success ratio for all speeds as was seen in figures 3.5 through 3.36. The effects of a hierarchical key management scheme can also be seen in these figures. Lowering the threshold level makes the certification success ratio higher, but this will increase availability. However, by decreasing the threshold, the level of security provided by the key management is decreased. Another advantage is to allow nodes to categorize their security needs. In previous work, only one threshold has been provided to the nodes; however not all applications require the same level of security. By varying the specific number of PCA nodes, we divided the security services provided by our key management technique into 5 different levels. Each level requires a different specific number of partial certificates to obtain a signature. The requester node can adjust its security needs according to the specific applications used i.e. sending e-mails vs. bank transactions. This is the most important advantage of our technique over similar solutions [4, 5, 6, 7, 8, 9, 10, 11].

Envisioning MANETs to be completely infrastructureless would be an extreme case. We strongly believe that MANETs will make use of the existing wireline network infrastructure where possible. Accordingly, the prevalence of the use of PKI in existing wireline infrastructure means that the ability of MANETs to also use PKI will better enable these networks to work together.

4.1 Future Work

In this thesis we presented a key management technique that supports distributed, fault tolerant and hierarchical security service availability for nodes for MANETs. In this work,

we showed that our proposed model performs well at the packet level. To get more precise results, one should implement our proposed solution on an actual test bed. The results will be more realistic and will provide an opportunity to see the actual cryptographic mechanism working.

We assume that there will be a certain node density in the field during initialization of the first specific number of shares (m_t). To initialize these first m_t PCA nodes, we assume an initializer who knows the full certificate signing key and associated polynomial $f(x)$ of degree $m_t - 1$. We assume a mobile node that is requesting certification services will have at least m_t initialized PCA nodes. However, due to high mobility, this may not always hold. Our hierarchical approach minimizes the possibility of getting less than m_t initialized PCA nodes. But, there might be a slight chance, up to a maximum 8%, of certificate request failure. A detailed analysis of the initialization process is an area for future study.

Authentication is another problematic point. When a new node joins the system, we assume that the node already has an initial certificate. In essence, the issuance of initial certificates is the problem of registering users. Initial certificates can be obtained in two ways:

1) The node may be issued an initial certificate by an off-line authority, after the authority verifies the authentication through other means, e.g. personal ID.
2) Any coalition of m_t networking PCA nodes may issue the initial certificate via collaborative admission control for this new node.

Further research into these authentication methods is required when designing routing algorithms. Most researchers ignore security needs for their protocols. Thus most routing protocols proposed in the literature lack reliable security mechanisms. Securing mature routing algorithms provides a potential direction for future research endeavors.

References

[1] C. Perkins, *Ad-Hoc Networking*. Addison-Wesley, USA, 2001.

[2] M. Ilyas, *The Handbook of Ad-Hoc Wireless Networks*. CRC Press, Florida, 2003.

[3] V. Karpijoki, "Security in ad-hoc networks," Helsinki University of Technology, Tech. Rep., 1999.

[4] W. Fumy and P. Landrock, "Principles of key management," *IEEE Journal Selected Areas Commun.*, vol. 11, no. 5, pp. 785–793, Jun. 1993.

[5] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.

[6] H. Luo and S. Lu, "Ubiquitous and robust authentication services for ad hoc wireless networks," Department of Computer Science, University of California Los Angeles, Tech. Rep., 2000.

[7] S. Yi and R. Kravets, "Key management for heterogeneous ad hoc wireless networks," *Proc. IEEE International Conference on Network Protocols*, pp. 202–203, Nov. 2002.

[8] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," *Proc. IEEE International Conference on Network Protocols*, pp. 251–260, 2001.

[9] S. Yi and R. Kravets, "Key management for heterogeneous ad-hoc wireless networks," Department of Computer Science, University of Illinois at Urbana-Champaign, Tech. Rep., Jul. 2002.

[10] F. Stajano, "The resurrecting duckling what next?" *Lecture Notes in Computer Science, Springer Verlag*, vol. 2133, pp. 204–214, 2001.

[11] B. Lehane, L. Doyle, and D.O'Mahony, "Shared rsa key generation in a mobile ad-hoc network," *Proc. IEEE Conference on Military Communications*, Oct. 2003.

[12] S. Yi and R. Kravets, "MOCA: Mobile Certificate Authority for Wireless Ad-Hoc Networks," *Proc. Annual PKI Research Workshop*, pp. 65–79, Apr. 2003.

[13] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized public-key management for mobile ad-hoc networks," *Proc. IEEE Transactions on Mobile Computing*, vol. 2, no. 1, pp. 52–64, Jan./Mar. 2003.

[14] H. Luo, P. Zerfos, J. K. S. Lu, and L. Zhang, "Self-securing ad-hoc wireless networks," *Proc. IEEE Symposium on Computers and Communications*, pp. 567 – 577, Jul. 2002.

[15] C. L. Liu, *Introduction to Combinatorial Mathematics.* McGraw Hill, New York, 1968.

[16] S. Yi and R. Kravets, "Practical PKI for ad-hoc wireless networks," Department of Computer Science, University of Illinois at Urbana-Champaign, Tech. Rep., Aug. 2001.

[17] B. Schneier, *Applied Cryptography.* John Wiley and Sons, New York, 1996.

[18] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, *Handbook of Applied Cryptography.* CRC Press, New York, 1997.

[19] A. M. Odlyzko, "Public key cryptography," *AT&T Technical Journal*, pp. 17–23, 1994.

[20] A. Shamir, "How to share a secret," *Communications of the ACM*, vol. 22, no. 11, pp. 612–613, Nov. 1979.

[21] F. Stajano and R. Anderson, "The resurrecting duckling: Security issues in ad-hoc wireless networks," *Proc. International Workshop on Security Protocols, Lecture Notes in Computer Science*, vol. 1796, pp. 172–194, Apr. 1999.

[22] P. S. Gemmel, "An introduction to threshold cryptography," The Technical Newsletter of RSA Laboratories, pp. 7–11, Winter 1997.

[23] D. Stinson, *Cryptography Theory and Practice.* CRC Press Inc., Florida, 1995.

[24] D. E. R. Denning, *Cryptography and Data Security.* Addison-Wesley, Canada, Jan. 1983.

[25] A. Weimerskirch and G. Thonet, "A distributed light-weight authentication model for ad-hoc networks," *Proc. International Conference on Information Security and Cryptology*, pp. 341–354, Nov. 2001.

[26] K. Fokine, "Key management in ad-hoc networks," Master's thesis, Linkopings University, Sweden, Sep. 2002.

[27] A. D. Santis, Y. Desmedt, Y. Frankel, and M. Yung, "How to share a function securely," *Proc. ACM Symposium on the Theory of Computing*, pp. 522–533, May 1994.

[28] V. Shoup, "Practical threshold signatures," *Proc. Eurocrypt 2000*, pp. 207–221, May 2000.

[29] K. Fall and K. Varadhan, "The NS Manual," UC Berkeley, LBL, USC/ISI, and Xerox PARC, URL: http://www.isi.edu/nsnam/ns-documentation.html.

[30] S. R. Das, C. E. Perkins, and E. M. Royer, "Performance comparison of two on-demand routing protocols for ad-hoc networks," *Proc. IEEE INFOCOM Conference on Computer Communications*, pp. 3–12, Mar. 2000.