SECURING MOBILE AD HOC NETWORK

ROUTING PROTOCOLS

NG KENG SENG

NATIONAL UNIVERSITY OF SINGAPORE

2003

SECURING MOBILE AD HOC NETWORK ROUTING PROTOCOLS

NG KENG SENG

(B.Eng.(Hons.), UNIMAS)

A THESIS SUBMITTED

FOR THE DEGREE OF MASTER OF ENGINEERING

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

NATIONAL UNIVERSITY OF SINGAPORE

2003

Acknowledgements

In preparing of this thesis, I have been fortunate to receive valuable assistance, suggestion, and support from my supervisor, and friends. I greatly appreciate their generosity in devoting their time to help me with this research.

I would like to express my thanks to my supervisor Dr. Winston Seah Khoon Guan, who has guided me throughout my entire candidature as a Research Scholar at the Institute for Infocomm Research (I^{2}R). I sincerely appreciate his enormous, innumerable pieces of advice, and insightful criticism. In addition, thanks to the National University of Singapore (NUS) for the Research Scholarship.

Besides, I would also like to extend my greatest thanks to my mother, Tina Loh Yoke Ying, my sister, Sandy Ng Sim Yee, and my girlfriend, Frances Chia Foong Sin, for their caring and love.

A very special thanks go to my close circle of friends, especially, Alvin, Yang Luqing, Li Feng, Wu Wei, Cheng Jing, Gao Qing, Hu Hongjie, Bijay and Luo Haihong, for their support, help and encouragement.

Table of Contents

ACKNOWLEDGEMENTS i

TABLE OF CONTENTS ii

LIST OF FIGURES v

LIST OF TABLES vii

LIST OF FLOWCHARTS viii

SUMMARY ix

CHAPTER 1 MOBILE AD HOC NETWORKS 1

1.1 INTRODUCTION 1

1.2 CHARACTERISTICS AND CHALLENGES 2

1.2.1 Network Origin 4

1.2.2 Network Capabilities 5

1.2.3 Network Transiency 5

1.3 APPLICATIONS 5

1.3.1 Military Tactical Networks 6

1.3.2 Personal Area Networks 6

1.3.3 Sensor Networks 6

1.3.4 Collaborative Networking 7

1.3.5 Disaster Area Networks 7

1.4 MOTIVATION 8

1.5 OUR CONTRIBUTION 9

1.6 THESIS ORGANIZATION 9

CHAPTER 2 NETWORK SECURITY ISSUES 10

2.1 NETWORK SECURITY 10

2.1.1 Security Services 10

2.1.2 Security Attacks.. 11

2.1.3 Security Mechanisms 13

2.2 CRYPTOGRAPHY BACKGROUND 14

2.2.1 Symmetric Encryption 14

2.2.2 Public Key Encryption 15

2.2.3 Digital Signature... 17

2.2.4 Digital Certificate 18

2.2.5 Secret Sharing 19

2.3 SECURITY ISSUES IN AD HOC NETWORKS 22

CHAPTER 3 SECURITY IN MOBILE AD HOC NETWORKS 23

3.1 INTRODUCTION 23

3.2 RELATED WORK 25

3.2.1 Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks 25

3.2.2 Secure Routing for Mobile Ad hoc Networks........................... 27

3.2.3 Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. 28

3.2.4 Authenticated Routing for Ad Hoc Networks........................... 29

3.2.5 Secure Ad hoc On-Demand Distance Vector Routing 30

CHAPTER 4 THE SECURE ROUTING PROTOCOL (SEROP) FOR MOBILE AD HOC NETWORKS 33

4.1 OPERATION OF CHAMP 33

4.2 THE PROPOSED PROTOCOL - SEROP 36

4.2.1 Assumptions 36

4.2.2 Distribution of Key 37

4.2.3 Route Discovery Request 38

4.2.4 Route Reply 40

4.2.5 Intermediate Node Reply 41

4.2.6 Route Maintenance & Deletion 43

4.2.7 Key Revocation 45

4.3 REASONING LOGIC 46

4.4 APPRAISAL OF SEROP 51

CHAPTER 5 SIMULATIONS AND RESULTS 55

5.1 INTRODUCTION 55

5.2 PERFORMANCE EVALUATION 56

5.3 SIMULATION SETUP 57

5.4 SIMULATION RESULTS FOR BENIGN NETWORK 58

5.5 SIMULATION RESULTS FOR MALICIOUS NETWORK 76

5.6 SUMMARY 80

CHAPTER 6 CONCLUSIONS AND FUTURE WORK 81

6.1 CONCLUSION 81

6.2 LIMITATION 82

6.3 FUTURE WORK 82

BIBLIOGRAPHY 83

List of Figures

1.1 Wireless Ad hoc Network 1

2.1 Symmetric Encryption Scheme 14

2.2 Public Key Encryption Scheme 15

2.3 Diffie-Hellman Key Exchange 16

2.4 Example of a Digital Signature 18

2.5 X.509 Certificate Format 19

4.1 Route Discovery Request 39

4.2 Route Reply 42

4.3 Intermediate Node Reply 44

4.4 Route Maintenance & Deletion 46

4.5 Overall Picture of the Protocol 47

4.6 The Protocol is Observed as an Exchange of Two Control Messages 48

5.1a Packet Delivery Ratio vs Pause Time 59

5.1b Packet Delivery Ratio vs Pause Time 60

5.2a Packet Delivery Ratio vs Number of Sources 61

5.2b Packet Delivery Ratio vs Number of Sources 62

5.2c Packet Delivery Ratio vs Number of Sources 63

5.2d Packet Delivery Ratio vs Number of Sources 64

5.3a End-to-end Delay vs Pause Time 65

5.3b End-to-end Delay vs Pause Time 66

5.4a End-to-end Delay vs Number of Sources 67

5.4b End-to-end Delay vs Number of Sources 68

5.4c End-to-end Delay vs Number of Sources 69

5.4d End-to-end Delay vs Number of Sources 70

5.5a Normalized Routing Overhead vs Pause Time 71

5.5b Normalized Routing Overhead vs Pause Time 72

5.6a Normalized Routing Overhead vs Number of Sources................................ 73

5.6b Normalized Routing Overhead vs Number of Sources................................ 74

5.6c Normalized Routing Overhead vs Number of Sources................................ 75

5.7 Packet Delivery Ratio vs Number of Malicious Nodes (Malicious Route Request) 77

5.8 Packet Delivery Ratio vs Number of Malicious Nodes (Malicious Route Reply).. 77

5.9 End-to-end Delay vs Number of Malicious Nodes (Malicious Route Request) 78

5.10 End-to-end Delay vs Number of Malicious Nodes (Malicious Route Reply) 79

5.11 Byte Overhead vs Number of Malicious Nodes (Malicious Route Request) 79

5.12 Byte Overhead vs Number of Malicious Nodes (Malicious Route Reply) .. 80

List of Tables

4.1 Table of Notation 37

4.2 Table of Statement 49

5.1 Simulation Parameters 58

List of Flowcharts

4.1 Route Discovery Request 40

4.2 Route Reply 41

4.3 Intermediate Node Reply 45

4.4 Route Maintenance & Deletion 46

Summary

Mobile Ad hoc Networks (MANETs) are made up of autonomous, collaborative mobile nodes with the ability to self-organize dynamically. Multi-hop routing capability is required for each node that wants to set up a connection with another node not in its radio range. An ad hoc network is essential in circumstances where the terrestrial restrictions require an entirely distributed network system without any permanent base stations.

Security is a critical issue for MANET, particularly for security-sensitive applications. However, security mechanisms for traditional networks are not wholly valid in MANET. Network operation can be easily jeopardized if the security schemes are not designed concurrently with the basic protocols. The spontaneous creation of the ad hoc networks makes it very difficult to differentiate between trusted and non-trusted nodes. An ad hoc network is dynamic because the nodes may leave and join the network anytime they wish. Consequently, the trust relationships between nodes change regularly as well. Any recommended security mechanisms with fixed arrangements would not be adequate. It is advantageous that the security solutions are capable of adapting on the fly to those changes and are scalable to handle a large network.

Our proposed protocol, Secure Routing Protocol (SEROP), is a hybrid cryptosystem that uses asymmetric key algorithm to establish secure routing between nodes and uses symmetric key algorithm to provide confidentiality of the data transmitted over the network. The protocol satisfies the fundamental aspects of

security like confidentiality, authentication, integrity, and non-repudiation. The protocol provides confidentiality not merely for routing information, but also for data messages. The originator of the message is able to encrypt the data packet with the secret key, which is shared among the sender and the intended recipients. The key exchange technique used in generating this shared secret key is based on the Diffie-Hellman key exchange. Reliance on this algorithm enables our protocol to approach forward secrecy.

SEROP presents a new approach to securing route discovery operation for mobile ad hoc network routing protocols. Any control message that does not correspond to the current pending request will be discarded by nodes along the source-destination path. The basis of our protocol is based on the broadly accepted route discovery processes by broadcasting route request packet, hence it enables our protocol to be an extension that can be easily adapted to reactive routing protocols.

For simulation implementation in both benign and malicious networks, we base our protocol, SEROP, on the basic operation of the Caching and Multi-Path (CHAMP) [1] routing protocol. The performance of SEROP in benign network is encouraging. It produces high packet delivery rate, low end-to-end delay and reasonable routing overhead. Besides that, the SEROP protocol is able to function and to perform well in the present of malicious nodes up to as high as 50%. SEROP can therefore be considered as an efficient and practical security extension that does not significantly degrade the overall performance of the based routing protocol.

Chapter 1

Mobile Ad hoc Networks

This chapter introduces Ad hoc networking and describes their salient features. It then discusses some of the challenging issues in these networks.

1.1 Introduction

Ad hoc networking is a networking paradigm for mobile, self-organizing networks. Typically, the network nodes are interconnected through wireless interfaces and unlike traditional networks lack specialized nodes, i.e. routers, that handle packet forwarding. Instead, every node in the network functions as a router as well as an application node and forwards packets on behalf of other nodes. Figure 1.1 shows such an example in which node A is not within reach of node C, however by using node B as an intermediate node, A and C are able to communicate.

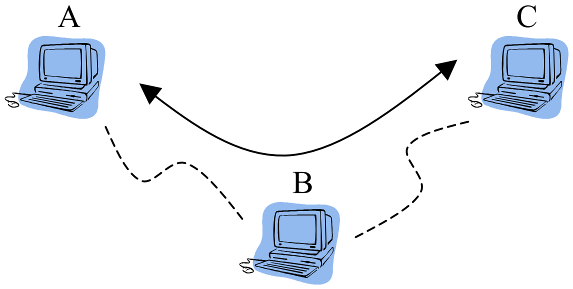


Figure 1.1 Wireless Ad hoc Network

Ad hoc networks have the ability to form *on the fly* and dynamically handle the joining or leaving of nodes in the network. For example, when three people with ad hoc networking enabled Personal Digital Assistants (PDAs) come within communication range of each other. The three PDAs can then automatically create an ad hoc network used to exchange data.

In many, if not most ad hoc networks, the nodes will also be mobile and they can then be termed mobile ad hoc networks, or MANET. The idea of MANETs is to incorporate routing functionality into mobile nodes in order to support robust and efficient operation in mobile wireless networks. Such networks are envisioned to have dynamic, rapid by changing, random, multi-hop topologies which are likely to be made up of relatively bandwidth-constrained wireless links. Multi-hop routing capability is essential for each node when the node needs to set up a connection with another node that is not in its radio range. The responsibility of network management is completely on the nodes even though they have unlimited mobility and connectivity. An ad hoc network is essential in circumstances where the terrestrial and geographical restrictions require an entirely distributed network system without any permanent base stations.

1.2 Characteristics and Challenges

When designing protocols for ad hoc networks, whether it be routing protocols or security protocols it is important to consider the characteristics of the network and realize that there are many flavours of ad hoc networks.

Mobile ad hoc networks generally have the following characteristics [5]:

- *Dynamic network topology*: The network nodes are mobile and thus the topology of the network may change frequently. Nodes may move around within the network but the network can also be partitioned into multiple smaller networks or be merged with other networks.

- *Limited bandwidth*: The use of wireless communication typically implies a lower bandwidth than that of traditional networks. This may limit the number and size of the messages sent during protocol execution.
- *Energy constrained nodes*: Nodes in ad hoc networks will most often rely on batteries as their power source. The use of computationally complex algorithms may not be possible. This also exposes the nodes to a new type of denial of service attack, the sleep deprivation torture attack [13], that aims at depleting the nodes energy.
- *Limited physical security*: The use of wireless communication and the exposure of the network nodes increase the possibility of attacks against the network. Due to the mobility of the nodes, the risk of them being physically compromised by theft, loss or other means will probably be greater than traditional network nodes.

In many cases, the nodes of the ad hoc network may also have limited CPU power and memory, e.g. low-end devices such as PDAs, cellular phones and embedded devices. As a result certain algorithms that are computationally or memory expensive might not be applicable.

Besides the characteristics mentioned above that are due to the nature of ad hoc networking the following aspects that depend more on the application should also be considered.

- *Network origin*: spontaneous vs. planned. Spontaneous: nodes with no prior relationship. Planned: nodes with a prior relationship, e.g. belonging to the same company, military unit etc.
- *Node capabilities*: uniform vs. diverse. Uniform: all nodes have approximately the same capabilities in terms of power source, CPU performance and memory size etc. Diverse: the nodes' capabilities differ significantly, certain nodes may be high-end computers while other are e.g. embedded devices.
- *Network transiency*: short term vs. long term. Short term: nodes come together to form an ad hoc network, and after the session has finished, no knowledge is retained about the other nodes. These networks typically only persist for a relatively short time period, i.e. less than a few hours. Long term: the same nodes will probably be part of the same ad hoc network multiple times and therefore save information about the other nodes for future use. These networks will persist for a longer time period. This also includes short-lived ad hoc networks that are created frequently.

Each of the aspects mentioned above will now be discussed with regard to how they affect the implementation of security services.

1.2.1 Network Origin

This aspect effects what assumptions or prerequisites that can be made on the nodes in the network. E.g. if it is a planned ad hoc network, it can perhaps be assumed that the nodes can be supplied with some initial data structures such as certificates, passwords, user names etc. However, if the network is spontaneous no such assumptions can be made.

1.2.2 Network Capabilities

If the capabilities of the nodes in the network are diverse, certain techniques may not be directly applicable. A certain technique may be applicable to a subset of the nodes but completely unusable on the rest of the nodes. An example of this could be the use of public key cryptography; although this is not an issue for high-end CPU's it may not be feasible for embedded devices.

1.2.3 Network Transiency

The longevity of the ad hoc network may influence the allowed complexity of some initialization phase. E.g. for a network consisting of nodes that will frequently join in an ad hoc network, it may be tolerable with a more complex initialization phase may be more tolerable than that of a network that will only last for a short time and will not recur.

1.3 Applications

To motivate the development of ad hoc networking protocols, there needs to be applications where the properties of ad hoc networking are beneficial. This section will discuss some such applications. Although some of these applications have been implemented many are still in the early research phase.

1.3.1 Military Tactical Networks

The first application of ad hoc networking was in the military domain. Ad hoc networking enables battlefield units to communicate anywhere and anytime, without the requirement of any fixed infrastructure. The fact that every node forwards packets also provides for a robust network. The loss of any one unit will not disrupt the network since there will be other units that can still provide packet forwarding services. Examples of military applications are the Tactical Internet [24] and the Saab NetDefence concept [25].

1.3.2 Personal Area Networks

The concept of personal area networks is about interconnecting different devices used by a single person, e.g. a PDA, cellular phone, laptop, and etc. In this case, the PDA or the laptop will connect with the cellular phone in an ad hoc fashion. The cellular phone can then be used to access Internet. Another example could be when a person holding a PDA comes within communication range of a printer. If both the PDA and the printer were ad hoc enabled, the PDA could automatically get access to the printing services.

1.3.3 Sensor Networks

Sensor networks [26] are ad hoc networks consisting of communication enabled sensor nodes. Each node contains one or more sensors, e.g. movement-, chemical- or heat sensors. When a sensor is activated, it relays the obtained information through the ad hoc network to some central processing node where further analysis and actions can be performed. Such sensor networks may consist of hundreds or thousands of sensors and can be used in both military and non-military applications, e.g. surveillance, environmental monitoring, etc. Sensor networks differ significantly from the other types of ad hoc networks described in this section. The most significant difference is the small size, extremely limited power resources and processing power of the sensor nodes.

1.3.4 Collaborative Networking

This application of ad hoc networking may be the most intuitive. The simplest example is when a group of people are attending a meeting and need to share information between their laptops or PDAs. If these devices were ad hoc enabled they could dynamically set up a network consisting of the meeting participants and thus enable the sharing of the information. Without ad hoc networking, a great deal of configuration and setup would be required to accomplish this task.

1.3.5 Disaster Area Networks

Ad hoc networking allows for the quick deployment of a communication network in areas where no fixed infrastructure is available or where the fixed infrastructure has been destroyed by natural disasters or other events. Thus, such networks could be used to improve the communication among rescue workers and other personnel and thereby support the relief efforts.

1.4 Motivation

MANET are characterized by decentralized network administration, i.e. each node acts both as host and router, and forwards packets for nodes that are not within transmission range of each other. Security in MANET is an essential component for basic network functions like packet forwarding and that network operation can be easily jeopardized if countermeasures are not embedded into basic network functions at the early stages of their design. Unlike networks using dedicated nodes to support basic functions like packet forwarding, routing, and network management, in ad hoc networks those functions are carried out by all available nodes. This very difference is the core of the security problems that are specific to ad hoc networks. As opposed to dedicated nodes of a classical network, the nodes of an ad hoc network cannot be trusted for the correct execution of critical network functions.

If an a priori trust relationship exists among the nodes of an ad hoc network, entity authentication would be sufficient to assure the correct execution of critical network functions. A priori trust can only exist in a few special scenarios like military networks and requires tamper-proof hardware for the implementation of critical functions. Entity authentication in a large network on the other hand raises key management requirements. If tamper-proof hardware and strong authentication infrastructure are not available, the reliability of basic functions like routing can be endangered by any node of an ad hoc network. No classical security mechanism can help counter a misbehaving node in this context. The correct operation of the network requires not only the correct execution of critical network functions by each participating node but it also requires that each node performs a fair share of the functions. The latter requirement seems to be a strong limitation for wireless mobile nodes whereby power saving is a major concern.

1.5 Our Contribution

The salient natures of ad hoc networks render them vulnerable to numerous types of security attacks. The dynamic feature of ad hoc networks makes enforcement of security an extremely challenging task. The main problem of ad hoc networks is that many proposed routing protocols are critical susceptibilities to security attacks. Effective operation of ad hoc networks depends on the maintenance of correct routing information of the network. Nevertheless, securing routing protocols without securing network transmissions is not enough. As a result, our major focuses are to secure the routing protocol and likewise to protect data transmission. In this thesis, we present the *Secure Routing Protocol (SEROP)* [10], which helps in achieving data confidentiality and securing the routing protocol for mobile ad hoc networks without demanding any unrealistic assumptions.

1.6 Thesis Organization

The thesis is structured into six chapters as follows. Chapter two discusses the security issues involved in networks. Chapter three provides an introduction to MANET security and presents related work. Chapter four presents the proposed Secure Routing Protocol (SEROP). Chapter five discusses the simulation results. Finally, we state our conclusions in chapter six after presenting the limitations and future possibilities of our work.

Chapter 2

Network Security Issues

This chapter provides the background information needed to understand the problems and the suggested solution. As our aim is to perform secure routing in ad hoc networks, we need to understand the security issues in a network.

2.1 Network Security

When discussing network security, three aspects can be covered; the services required, the potential attacks and the security mechanisms.

The security services aspect includes the functionality that is required to provide a secure networking environment while the security attacks cover the methods that could be employed to break these security services. Finally the security mechanisms are the basic building blocks used to provide the security services.

2.1.1 Security Services

In providing a secure networking environment, some or all of the following services may be required [19]:

- *Confidentiality*: Ensures that transmitted information can only be accessed by the intended receivers.
- *Authentication*: Allows the communicating parties to be assured of the others identity.

- *Integrity*: Ensures that the data has not been altered during transmission.
- *Non-repudiation*: Ensures that parties can prove the transmission or reception of information by another party, i.e. a party cannot falsely deny having received or sent certain data.
- *Availability*: Ensures that the intended network services are available to the intended parties when required.

Depending on the capabilities of any potential attacker, different mechanisms may be used to provide the services above.

2.1.2 Security Attacks

Security attacks can be classified in the following two categories [19] depending on the nature of the attacker:

- *Passive attacks*: The attacker can only eavesdrop or monitor the network traffic. Typically this is the easiest form of attack and can be performed without difficulty in many networking environments, e.g. broadcast type networks such as Ethernet and wireless networks.
- *Active attacks*: The attacker is not only able to listen to the transmission but is also able to actively alter or obstruct it.

Furthermore depending on the attackers' actions, the following subcategories can be used to cover the majority of attacks.

- *Eavesdropping*: This attack is used to gain knowledge of the transmitted data. This is a passive attack which is easily performed in many networking environments as mentioned above. However, this attack can be easily prevented by using an encryption scheme to protect the transmitted data.

- *Traffic analysis*: The main goal of this attack is not to gain direct knowledge about the transmitted data, but to extract information from the characteristics of the transmission, e.g. amount of data transmitted, identity of the communicating nodes etc. This information may allow the attacker to deduce sensitive information, e.g. the roles of the communicating nodes, their position etc. Unlike the previously described attack this one is more difficult to prevent.
- *Impersonation*: Here the attacker uses the identity of another node to gain unauthorized access to a resource or data. This attack is often used as a prerequisite to eavesdropping. By impersonating a legitimate node the attacker can try to gain access to the encryption key used to protect the transmitted data. Once this key is known by the attacker, the eavesdropping attack can be carried out.
- *Modification*: This attack modifies data during the transmission between the communicating nodes, implying that the communicating nodes do not share the same view of the transmitted data. An example could be when the transmitted data represents a financial transaction where the attacker has modified the transaction value.
- *Insertion*: This attack involves an unauthorized party, who inserts new data claiming that it originated from a legitimate party. This attack is related to that of impersonation.
- *Replay*: The attacker retransmits data previously transmitted by a legitimate node.

- *Denial of service*: This active attack aims at obstructing or limiting access to a certain resource. This resource could be a specific node, service or the whole network.

2.1.3 Security Mechanisms

Most of the security services previously mentioned can be provided using different cryptographic techniques. The following subsections give an overview of which techniques are used to provide each of the services.

- *Confidentiality:* The confidentiality service can be of two different types. The most common type of confidentiality requirement is that transmitted information should not be exposed to any unauthorized entities. A stricter confidentiality requirement is that the very existence of the information should not be revealed to any unauthorized entities. The first type of confidentiality requirement only requires protection from eavesdropping attacks and can be provided using an encryption scheme. The stricter requirement implies that the service must also provide protection against traffic analysis. Such a service will typically require additional mechanisms along with some encryption scheme.
- *Integrity:* The integrity service can be provided using cryptographic hash functions along with some form of encryption. When dealing with network security, the integrity service is often provided implicitly by the authentication service.
- *Authentication:* Authentication can be provided using encryption along with cryptographic hash functions.

- *Non-repudiation:* Non-repudiation requires the use of public key cryptography to provide digital signatures. Along with digital signatures a trusted third party must be involved.
- *Availability:* The availability is typically ensured by redundancy, physical protection and other non cryptographic means, e.g. use of robust protocols.

2.2 Cryptography Background

2.2.1 Symmetric Encryption

Symmetric encryption is illustrated in figure 2.1. The plain text message *m* is encrypted using the shared key *k*, resulting in the cipher text *c*. To recover the plain text message the cipher text is decrypted using the same key used for the encryption. Symmetric encryption schemes can be used to provide confidentiality, integrity and authentication. The shared key must be distributed over a secure communication channel.

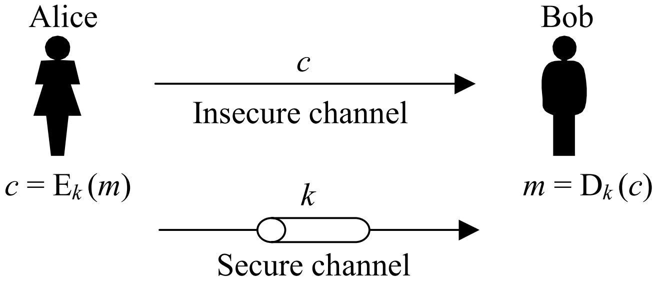


Figure 2.1 Symmetric Encryption Scheme

2.2.2 Public Key Encryption

Unlike conventional encryption schemes where the involved parties share a common encryption/decryption key, public key encryption schemes depend on the use of two different but mathematically related keys. One of the keys is used for encryption and the other for decryption. The public key encryption scheme is illustrated in figure 2.2. Bob generates a pair of keys, his public/private key pair pk_{Bob}/sk_{Bob}. The public key is related to the private key, but in such a way that the private key cannot be derived from it without additional information.

If Alice wants to send an encrypted message to Bob, she first needs to obtain his public key. As the name implies Bob's public key does not need to be kept secret, however it must be authenticated, i.e. Alice must be assured that the public key she believes belongs to Bob is really his.

Once Alice has Bob's authentic public key pk_{Bob}, she encrypts the plain text message m using it. The resulting cipher text c can then only be decrypted using Bob's private key sk_{Bob} which only Bob knows.

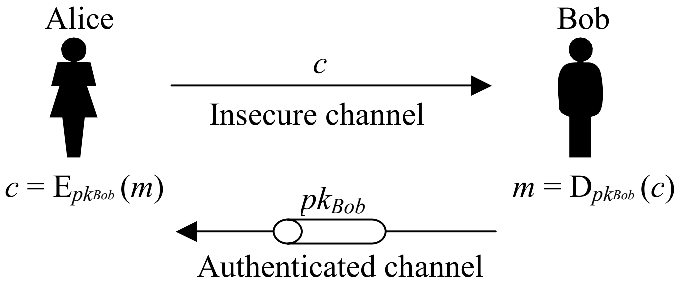


Figure 2.2 Public Key Encryption Scheme

Compared with symmetric encryption, public key encryption has a less stringent requirement for the communication channel over which the key distribution is performed. Public key encryption only requires an authenticated channel as opposed to a secure channel that is required for the distribution of symmetric encryption keys.

Public key encryption can also provide non-repudiation along with confidentiality, integrity and authentication. However, public key encryption requires much more computational resources than symmetric encryption and therefore has much lower performance. Consequently, public key encryption is typically used to encrypt only small amounts of data, e.g. symmetric encryption keys and digital signatures.

Diffie-Hellman

The Diffie-Hellman (DH) algorithm was the first public key algorithm published. However, it is limited to securely exchanging keys that can subsequently be used to provide the security services mentioned above.

The DH algorithm, illustrated in figure 2.3, requires two public parameters, a prime p and a generator g of Z_p. A generator of Z_p *is* an integer g such that $g, g^2, ..., g^{p-1}(mod\ p)$ generate the values 1 through $p - 1$ in some order. To exchange a shared key Alice and Bob generate the random secrets x_{Alice} and x_{Bob}. Bob then sends $y_{Bob} = g^{X_{Bob}}$ $mod\ p$ to Alice and Alice sends $y_{Alice} = g^{X_{Alice}}\ mod\ p$ to Bob. Alice and Bob can then generate the shared secret key k as:

$$k = (y_{Alice})^{X_{Bob}} = (y_{Bob})^{X_{Alice}} = g^{X_{Bob}.X_{Alice}}(mod\ p)$$

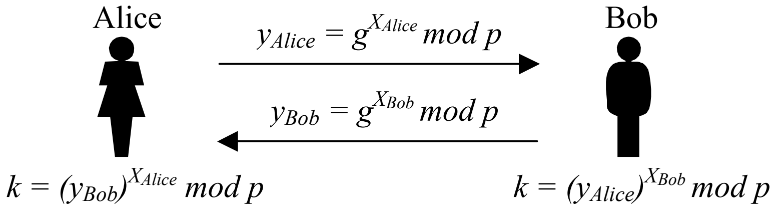

Alice
Bob
$y_{Alice} = g^{X_{Alice}}\ mod\ p$
$y_{Bob} = g^{X_{Bob}}\ mod\ p$
$k = (y_{Bob})^{X_{Alice}}\ mod\ p$
$k = (y_{Alice})^{X_{Bob}}\ mod\ p$

Figure 2.3 Diffie-Hellman Key Exchange

RSA

RSA is a public key encryption algorithm that can be used to provide confidentiality, integrity, authentication and non-repudiation services. To encrypt a message *m* or decrypt a cipher text *c*, the following calculations are performed:

$$c = m^e \, mod \, n$$

$$m= c^d \, mod \, n = m^{ed} \, mod \, n$$

If the algorithm is intended to be used to provide confidentiality, the values *n* and *e* are made publicly known while *d* is kept secret, viz the public key $pk = \{e, n\}$ and the private key $sk = \{d, n\}$. For user A to encrypt a message intended for user B, B's public key pk_B is used for the encryption, $c = Epk_B\,(m) = m^e \, mod \, n$. Since only B has knowledge of the secret key sk_B it alone can decrypt the cipher text and recover the plain text, $m = Dsk_B\,(c) = c^d \, mod \, n$.

2.2.3 Digital Signature

A digital signature is a data structure that provides proof of origin, i.e. authentication and integrity, and depending on how it is used, it can also provide non-repudiation. Figure 2.4 illustrates how a digital signature is used. Alice wants to send a message to Bob, however she does not want it to be modified during transmission and Bob wants to be sure that the message really came from Alice. What Alice does is that she signs on the digest of the message using her private key sk_{Alice}. She then sends both the message and the signed digest which is her signature. Bob can then verify the signature by computing the hash digest of the message he received and comparing it with the digest he gets when verifying the signature using Alice's public key pk_{Alice}. If the digests are equal, Bob knows that Alice sent the message and that it has not been modified since she signed it.

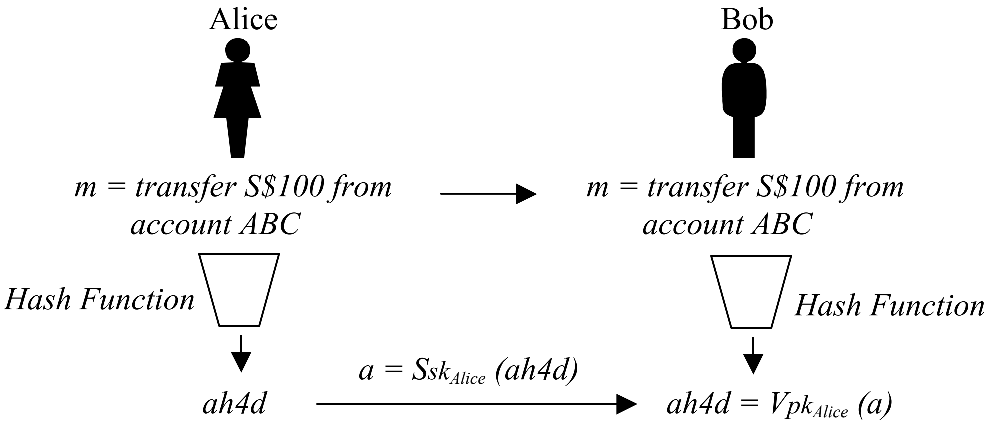


Figure 2.4 Example of a Digital Signature

2.2.4 Digital Certificate

Public key cryptography is very useful, but in the presence of active attackers a problem arises. Consider the following, Alice wants to send a secret message to Bob, so she encrypts the message using Bobs public key pk_{Bob} that she has retrieved from a server. However, the key that Alice retrieved actually belongs to an attacker. The secret message which was intended for Bob can now be decrypted and read by the attacker. Digital certificates are used to prevent this type of attack.

Basically a digital certificate is a statement issued by some trusted party saying that it verifies that the public key pk_A in fact belongs to the user A. The trusted party digitally signs this statement and therefore anyone with the authentic public key of the trusted party can verify the certificate and thereafter use pk_A and be sufficiently sure that it actually belongs to node A.

Figure 2.5 shows the information in an X.509 certificate. The serial number is used to uniquely identify the certificate, and issuer name is the name of the trusted party who has issued the certificate. The validity field specifies how long the certificate is valid. The subject is the entity being identified by the certificate, i.e. the entity whose public key is being certified. The next two fields contain the public key being certified and information about what it is certified to be used for (e.g. encryption, signatures etc.). The extensions field can be used to specify any additional information about the certificate. The signature field contains the certificates signature along with information about the hash algorithm used etc.

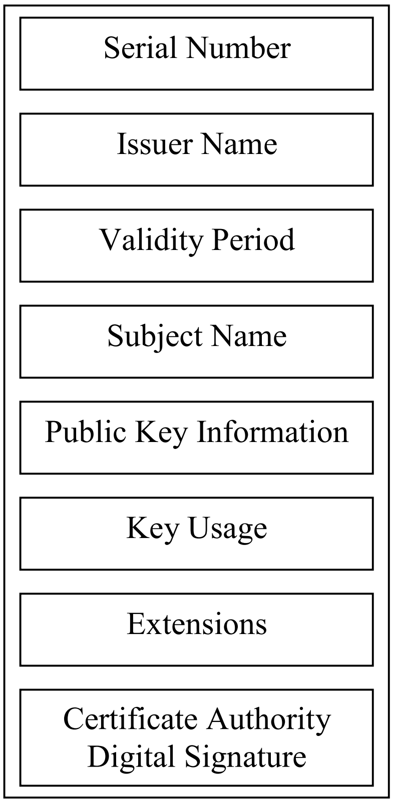


Figure 2.5 X.509 Certificate Format

2.2.5 Secret Sharing

Secret sharing allows a secret to be shared among a group of users (share holders) in such a way that no single user can deduce the secret from his share alone. Only by combining (a sufficient number of) the shares can the secret be reconstructed. A secret sharing scheme where *k* out of *n* share holders are needed to reconstruct the secret is referred to as a *(k, n)* threshold scheme.

Shamir's Secret Sharing

This *(k, n)* threshold secret sharing scheme proposed by Adi Shamir [27] is based on polynomial interpolation and works as follows. The secret S is to be shared among the n shareholders identified by id_i, $i = 1, \ldots, n$. The following steps are performed by the dealer who is the trusted party responsible for generating the secret and distributing it to the users:

1. A prime p is chosen such that $p > max\ (S, n)$.
2. A polynomial $f(x) = a_0 + a_1x + \ldots + a_{k-1}x^{k-1}$ is generated where $a_0 = S$ and a_i, $i = 1, \ldots, k-1$ are chosen randomly from Z_p.
3. The shares S_i, $i = 1, \ldots, n$ are generated as $S_i = f(id_i)(mod\ p)$.
4. The shares are securely distributed to the respective shareholders.

To reconstruct the secret Lagrange interpolation is used. With the knowledge of a minimum of k shares the polynomial $f(x)$ can be reconstructed and the secret recovered by calculating $f(0)$. The Lagrange interpolation is described below:

$$f(x) = \sum_{i=1}^{k} S_i . l_{id_i}(x)(\text{mod}\, p) \text{ where } l_{id_i}(x) = \prod_{j=1, j \neq i}^{k} \frac{x - id_j}{id_i - id_j}$$

It is important that no shareholder gains knowledge of any share other than his own. Otherwise he could potentially gain knowledge of k shares and then be able to reconstruct the secret himself. Therefore a trusted party is needed to perform the reconstruction of the secret, i.e. the shareholders provide their shares to the trusted party who performs the action requiring the secret, e.g. the signing of certificates etc.

Proactive Secret Sharing

In the secret sharing scheme described above the secret is protected by distributing it among several shareholders. However, given sufficiently long time an attacker could compromise *k* shareholders and obtain their shares, thereby allowing him to reconstruct the secret. To defend against such attackers proactive secret sharing schemes update the shares on a regular basis. An attacker must then compromise *k* shareholders between the updates since only *k* shares belonging to the same update period can be used to reconstruct the secret.

The share update is achieved by adding a random update polynomial $f_{update}(x)$ to the original sharing polynomial *f(x)* as follows:

$$f(x) = a_0 + a_1x + ... + a_{k-1}x^{k-1} (\bmod p),$$

$$f_{update}(x) = b_1x + b_2x^2 + ... + b_{k-1}x^{k-1} (\bmod p),$$

$$\begin{aligned} f_{new}(x) &= f(x) + f_{update}(x) \\ &= a_0 + (a_1 + b_1)x + ... + (a_{k-1} + b_{k-1})x^{k-1} (\bmod p) \end{aligned}$$

The update shares $S_{i,updated}$ can then be calculated as $f_{new}(id_i)$ where $i = 1,...,k$. However, in practice it is enough to calculate the share of the update polynomial, $\overline{S}_i i = 1,...,k$ and securely distribute them to the respective shareholders. Each shareholder then adds it to it original share to obtain the updated share, i.e. $S_{i,updated} = S_i + \overline{S}_i (\bmod p)$.

Verifiable Secret Sharing

If any shareholder wishes to prevent the reconstruction of the secret, he can provide an invalid share, e.g. a random value, to be used for the reconstruction. The Lagrange interpolation will then result in the reconstruction of a value $\hat{S}$, different from the secret S. Verifiable secret sharing mechanisms are used to prevent this type of denial of service attack.

The mechanism works as follows:

1. Prior to distributing the shares to the shareholders, the dealer publishes $g^{a(0)}, g^{a(1)}, \ldots, g^{a(k-1)}$ that are witnesses of the coefficients of the sharing polynomial.
2. Each node can then upon receiving its share verify it by checking that $g^{S(i)} = g^{a(0)} \cdot (g^{a(1)})^{id(i)} \cdot \ldots \cdot (g^{a(k-1)})^{id(i)\wedge(k-1)}$

2.3 Security Issues in Ad hoc Networks

The nature of ad hoc networks makes them vulnerable to various forms of attack. Wireless networks are typically much easier to snoop on as only physical proximity is required to gain access to the medium. The impromptu nature of the ad hoc network formation makes it hard to distinguish between trusted and non-trusted nodes. In the most general form, nodes may leave and join the network at will. Due to the dynamic nature of ad hoc networks, the trust relationship between nodes also changes. Any security solution with static configuration would not suffice. It is desirable that the security mechanisms adapt on-the-fly to those changes. They should also be scalable to handle large networks. The random nature of these networks makes enforcement of security a challenging issue.

Chapter 3

Security in Mobile Ad hoc Networks

This chapter provides an introduction to MANET security. It then briefly discusses the related works.

3.1 Introduction

Research in security for ad hoc networks is in the early stages of development. With relatively few security schemes proposed for ad hoc networks, threshold schemes have been frequently suggested for improving the security of mobile networks where the nodes have relatively poor physical protection. Zhou and Haas [12] have proposed the use of threshold scheme as a mechanism for rendering security to the network. Nevertheless, threshold scheme can effortlessly lead to the denial of service although it obviously improves the security of the system. Denial of service attacks essentially intimidates the operation in all kinds of networks and it is unfeasible to prevent this type of attack in general. Prompt decision-making and corrective-action are often more important than protection against compromised nodes in the practical implementation. A solution for increasing route robustness for the networks is the utilization of redundant paths, as stated in [12]. The effectiveness of this protection is restricted because the end point of a route is not always capable of discovering the attack by the malicious node.

The basic security role for ad hoc networks is to set up a secure communications channel between the participating nodes. This can be done by generating a shared secret key for the encryption and authentication of the data packet to be sent between the nodes. An efficient protocol for key exchange has been presented in [8]. In [15], the authors suggest a password-based authentication protocol that is derived from the Encrypted Key Exchange protocol, with emphasis on the robustness of the protocol against the failure of some nodes. In this case, the secure connections between the participants are created from a manually exchanged password. Hence, no support infrastructure is needed. It should be noted that a shared secret key is unable to prevent the group members from eavesdropping on each other. The idea is that the member nodes should trust each other with respect to the purpose of the group.

A small number of security schemes for internal attacks have been proposed for ad hoc networks. An architecture for cooperative and distributed intrusion detection for wireless ad hoc networks is presented in [18]. The primary assumption is that the user and program activities are observable. The process of building an anomaly detection model is discussed. The major drawback of anomaly detection is that it may not be able to describe what the attack is and may have high false positive rate. Other proposals for intrusion detection which introduce techniques that improve throughput by identifying misbehaving nodes in an ad hoc network [17] and proposed the intrusion detection agent to prevent some internal attacks on network [11].

Most of the MANET routing protocols can deal with the dynamically changing nature of the ad hoc networks. However, none of these protocols appear to be able to handle security appropriately. The majority of the routing protocols do not take into account the necessary security needs at the present time [9]. This leads to the presumption that security mechanisms will be retrofitted after the proposed protocol has been tested well enough. Consequently, it might lead to unforeseeable and untraceable vulnerabilities in the system if the security schemes are not designed concurrently with the basic routing protocol. However, all the MANET routing protocol proposals do not disregard the security issues entirely. Some of the routing protocols suggest and believe that IPSec [4] is able to provide excellent confidentiality, authentication and protection mechanism so that the security issues need not be handled by the routing protocol itself. Retrofitting IPSec to the existing routing protocol, however, would produce additional overheads to the system.

Furthermore, IPSec does not secure the routing protocol; it only provides security and authentication between two end nodes with existing routes to each other. Some of the security schemes proposed for on-demand routing protocols rely on symmetric cryptography [22,23] which assume the existence of the security association and shared secret keys between source and destination node. Such assumptions ignore the key distribution mechanisms which could increase the overheads in the network and decrease the performance of the protocol.

3.2 Related Work

3.2.1 Secure Efficient Distance Vector Routing in Mobile Wireless Ad Hoc Networks

Secure Efficient Ad hoc Distance vector routing protocol (SEAD) [20] is robust against multiple uncoordinated attackers creating incorrect routing state in any other node, even against active attackers or compromised nodes in the network. The design of SEAD is based in part on the Destination-Sequenced Distance-Vector ad hoc network routing protocol (DSDV) [7]. In order to support the use of SEAD with nodes of limited CPU processing capability, and to guard against Denial-of-Service attacks in which an attacker attempts to cause other nodes to consume excess network bandwidth or processing time, SEAD uses efficient one-way hash functions.

A one-way hash chain is built on a one-way hash function. To create a one-way hash chain, a node chooses a random $x \in \{0,1\}^{\rho}$ and computes the list of values $h_0, h_1, h_2, h_3, \ldots, h_n$ where $h_0 = x$, and $h_i = H(h_{i-1})$ for $0 < i \leq n$, for some n. The node at initialization generates the elements of its hash chain as shown above, from left to right and then over time uses certain elements of the chain to secure its routing updates; in using these values, the node progresses from right to left within the generated chain.

Each node in SEAD uses a specific single next element from its hash chain in each routing update that it sends about itself (metric 0). Based on this initial element, the one-way hash chain conceptually provides authentication for the lower bound of the metric in other routing updates for this destination; the authentication provides only a lower bound on the metric: an attacker can increase the metric, claim the same metric, but cannot decrease the metric.

The method used by SEAD for authenticating an entry in a routing update uses the sequence number in that entry to determine a contiguous group of m elements from that destination node's hash chain, one element of which must be used to authenticate that routing update. The particular element from this group of elements that must be used to authenticate the entry is determined by the metric value being sent in that entry.

3.2.2 Secure Routing for Mobile Ad hoc Networks

The provision of comprehensive secure communication for mobile ad hoc networks mandates that both route discovery and data forwarding be safeguarded. The Secure Routing Protocol (SRP) [22] counters malicious behavior that targets the discovery of topological information. Protection of data transmission is addressed through the related Secure Message Transmission Protocol (SMT), which provides a flexible, end-to-end secure data forwarding scheme that naturally complement SRP.

SRP provides correct routing information, i.e., factual, up-to-date, and authentic connectivity information regarding a pair of nodes that wish to communicate in a secure manner. The sole requirement is that any two such *end* nodes have a security association. Accordingly, SRP does not require any of the intermediate nodes to perform cryptographic operations or have a prior association with the end nodes. As a result, its end-to-end operation allows for cryptographic mechanisms, such as message authentication codes.

SRP discovers one or more routes whose correctness can be verified from the route *geometry* itself. Route requests propagate verifiably to the sought, trusted destination. Route replies are returned strictly over the reversed route, as accumulated in the route request packet. In order to guarantee this crucially important functionality, the interaction of the protocol with the IP-related functionality is explicitly defined.

It has been shown that, over a range of scenarios, SRP is successful in providing correct routing information in a timely manner. It can do so even in the presence of adversaries that disrupt the route discovery. Moreover, the observation shows that the processing overhead due to cryptographic operations remains low, allowing the protocol to remain competitive to reactive protocols, which do not incorporate security features at all.

3.2.3 Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks

Ariadne [23] is a secure on demand routing protocol that relies on symmetric cryptography. Ariadne can authenticate routing messages using one of three schemes: shared secrets between each pair of nodes, shared secrets between communicating nodes combined with broadcast authentication, or digital signatures. The use of Ariadne with TESLA (Time Efficient Stream Loss-tolerant Authentication) [6], an efficient broadcast authentication scheme that requires loose time synchronization, is discussed primarily in this section. Using pairwise shared keys avoids the need for synchronization, but at the cost of higher key setup overhead.

Basic Ariadne Route Discovery: The design of the Ariadne protocol is presented in two stages: a mechanism that enables the target to verify the authenticity of the ROUTE REQUEST is presented first, and follow by an efficient per-hop hashing technique to verify that no node is missing from the node list in the REQUEST. The initiator **S** performs a Route Discovery for target **D** is assumed in the following discussion, and that they share the secret keys K_{SD} and K_{DS}, respectively, for message authentication in each direction.

Target authenticates ROUTE REQUEST*s.* To convince the target of the legitimacy of each field in a ROUTE REQUEST, the initiator simply includes a MAC computed with key K_{SD} over unique data, for example a timestamp. The target can easily verify the authenticity and freshness of the route request using the shared key K_{SD}.

In a Route Discovery, the initiator wants to authenticate each individual node in the node list of the ROUTE REPLY. A secondary requirement is that the target can authenticate each node in the node list of the ROUTE REQUEST, so that it will return a

ROUTE REPLY only along paths that contain legitimate nodes. Each hop authenticates new information in the REQUEST. The target buffers the REPLY until intermediate nodes can release the corresponding TESLA keys. The TESLA security condition is verified at the target, and the target includes a MAC in the REPLY to certify that the security condition was met.

Per-hop hashing. Authentication of data in routing messages is not sufficient, as an attacker could remove a node from the node list in a REQUEST. One-way hash functions are used to verify that no hop was omitted, and this approach is called *per-hop hashing*. To change or remove a previous hop, an attacker must either hear a REQUEST without that node listed, or must be able to invert the one-way hash function.

Basic Ariadne Route Maintenance: Route Maintenance in Ariadne is based on DSR. A node forwarding a packet to the next hop along the source route returns a ROUTE ERROR to the original sender of the packet if it is unable to deliver the packet to the next hop after a limited number of retransmission attempts. To prevent unauthorized nodes from sending ERRORs, an ERROR is required to be authenticated by the sender. Each node on the return path to the source forwards the ERROR. If the authentication is delayed, for example when TESLA is used, each node that will be able to authenticate the ERROR buffers it until it can be authenticated.

Avoiding Routing Misbehavior: The protocol described so far is vulnerable to an attacker that happens to be along the discovered route. In particular, Ariadne has not presented a means of determining whether intermediate nodes are in fact forwarding packets that they have been requested to forward. Routes based on their prior performance in packet delivery are chosen. The scheme relies on feedback about which packets were successfully delivered. The feedback can be received either through an extra end-to-end network layer message, or by exploiting properties of transport layers, such as TCP with SACK [56]; this feedback approach is somewhat similar that used in IPv6 for Neighbor Unreachability Detection [21].

A node with multiple routes to a single destination can assign a fraction of packets that it originates to be sent along each route. When a substantially smaller fraction of packets sent along any particular route are successfully delivered, the node can begin sending a smaller fraction of its overall packets to that destination along that route.

3.2.4 Authenticated Routing for Ad Hoc Networks

The Authenticated Routing for Ad hoc Networks (ARAN) protocol [2] uses public key cryptography to guarantee message authentication, integrity and non-repudiation. The protocol is designed for the *managed-open* environment, where nodes can obtain a public key certificate from a common certification authority that is trusted by all other nodes in the environment. Typical examples of such an environment are classroom or conference scenarios. The operation of the protocol can be divided into route discovery and route maintenance phases.

The route discovery process is initiated by the source node by flooding a digitally signed Route Discovery packet (RDP) to its neighbors. When a neighbor *A* receives the RDP message, it sets up a reverse path back to the source node and verifies the signature of the source by extracting *S*'s public key from its certificate. The node then signs the contents of the message, appends its own certificate, and broadcasts the message to its neighbors. When *A*'s neighbor *B* receives the message, it validates *A*'s signature, and then replaces it with its own signature (the signature of the source node is retained). The packet continues to be rebroadcast in this manner across the network until it reaches the destination.

When the first RDP reaches the destination, the destination node verifies the signature of the source node and then sends a digitally signed Route Reply packet (REP) back to the source. The REP travels along the same path as the RDP, and the same signing procedure is performed by intermediate nodes. Note that because the destination must sign the REP message, only the destination is allowed to respond to the RDP. Also, because RDP messages are signed at each hop and do not contain a hop count or a source route, malicious nodes have no opportunity to intentionally redirect traffic.

Route maintenance is performed through digitally signed Error messages that are initiated by the node directly upstream of a link failure.

3.2.5 Secure Ad hoc On-Demand Distance Vector Routing

The Ad hoc On-Demand Distance-Vector (AODV) [3] routing protocol is a reactive routing protocol for mobile nodes in ad hoc networks. The Secure AODV (SAODV) [14] routing protocol is an extension of the AODV routing protocol that provides security features for the route discovery mechanism.

The fundamental idea of this scheme is that the original sender of the routing message attaches a signature to the AODV packet using their private key. The destination node or intermediate node can generate the route replies. However, an intermediate node that wants to reply a request needs not only the correct route, but also the signature corresponding to that route to add it in the route reply and the lifetime that came in the same message of the signature. The hop count of all the control messages is verified using a hash chain.

The SAODV scheme assumes that each node is able to obtain the public keys of others network nodes and public key of the certification authority if the nodes connect periodically to a fixed network. This assumption is unreasonable because it ignores the public key distribution which could increase the traffic load in the networks. Applying hash chains for authenticating the hop counter could lead to a problem because a malicious node might not increment the hop counter and make use of the same hash value when forwarding a route.

Chapter 4

The Secure Routing Protocol (SEROP) for Mobile Ad hoc Networks

The Secure Routing Protocol, SEROP works as an extension to an on-demand routing protocol for mobile ad hoc networks. For the simulation purposes, we base our protocol, SEROP, on the basic operation of the Caching and Multi-Path (CHAMP) [1] routing protocol.

4.1 Operation of CHAMP

CHAMP is a reactive routing protocol for mobile ad hoc networks that utilizes data caching and shortest multiple path routing to support mobility and achieve energy-efficiency. In this section, we briefly describe the basic operation of CHAMP.

CHAMP uses three kinds of control messages, namely route request (RREQ), route reply (RREP) and route error (RERR). Every node in the network keeps two data structures, namely route cache and route request cache. Route cache stores next hop information and other details required for data packet forwarding. Meanwhile, the information of recently received and processed route requests is stored in the route request cache. Besides that, each node also keeps a FIFO send buffer for storing packets waiting for routes and a data cache for storing recently forwarded data packets.

A data packet is recognized by the source identifier and a sequence number. CHAMP picks the least used route for the transmission of data packets in order to balance the load. In general, when a node has data to forward, if there is more than one route to the destination in its route cache, it chooses the route with the least use count and forwards the data packet. Then, it keeps a copy of the data packet in its data cache. However, if there is no route to the destination, it saves the data in its send buffer and carries out a route discovery instantly.

Route discovery is started by a source node that has a data packet to send but has no available route to the intended destination by broadcasting a RREQ packet. In order to ensure that the protocol is robust against topology changes, a node is encouraged to discover multiple routes to a particular destination. However, to prevent network congestion, sending of the request is separated by an increasing interval using binary exponential back off.

When a node receives an RREQ and it has no active route to the destination, it records the previous hop and forward count of the packet. The previous hop nodes that send requests with the lowest forward count are included in the RREP receivers set of the node. This receiver set is used in the creation of a route reply packet. However, if the node has an active route to the destination, it can reply to the request provided that its distance to the destination is less than the last hop count encoded in the request.

When the destination node receives a RREQ from its neighbour, it immediately sends back a RREP if the hop count is less than the minimum forward count. RREP contains a set of nodes that can receive it which taken from the RREP receivers set and the set nodes that forward the same RREQ.

Route maintenance occurs only when all active routes fail. A local route repair is performed by carrying out a limited route search. If the repair fails, a route deletion is performed by notifying the upstream nodes to remove that particular route and reroute the affected data. Since the data packets are cached, an upstream node with an alternate route is able to retransmit the same data again to the destination. This leads to a small routing overhead and savings in energy consumption.

Extensive simulations [1] have been performed to gauge the performance of CHAMP and compare it with AODV and DSR. Simulation results show that by using a five-packet data cache and two routes per destination configuration, CHAMP is able to achieve good improvement in packet delivery, outperforming AODV and DSR by up to 30% in very congested scenarios and the delay of CHAMP is half that of AODV and DSR. In terms of routing overhead, CHAMP generates a relatively lower overhead at higher mobility rates. Based on these finding, we can assume that implementing SEROP over CHAMP will generally give better performance than implementing it over AODV and DSR.

4.2 The Proposed Protocol – SEROP

Security is a crucial issue in any network. The dynamic feature of ad hoc networks makes it very difficult to guarantee secure communication in these networks. Effective performance of ad hoc networks relies on the maintenance of proper network routing information. Nevertheless, securing routing protocols without securing network transmissions is not adequate. Therefore, we focus on protecting the data transmission and making the routing protocol secure. In this section, we present the *Se*cure *Ro*uting *P*rotocol (SEROP) [10] for mobile ad hoc networks that achieves data confidentiality and secures the routing protocols.

4.2.1 Assumptions

The formation of the network is accomplished after the approval of the Master node ***M*** that created the network. We assume that the network comprises of a group of mutually trusting nodes. All the links between the nodes are bi-directional. In addition, all nodes are capable of carrying out the encryption algorithms with limited computational power. The adversary lacks the computational power to break the protocol that we have designed through brute force methods. All nodes in the network trust any data message signed using the corresponding private key. Finally, each node is assumed to have enough memory to store information such as the public keys of other nodes.

4.2.2 Key Distribution

SEROP is a hybrid cryptosystem. Public-key algorithm is used to establish secure routing between nodes, and symmetric key algorithm provides confidentiality of the data transmitted over the network. Table 4.1 summarizes our notations. The network creator is the only entity that has the master public and private key pairs. Before entering a network, each node needs to send its own public key to the master node in order to get the certificate. Besides that, each node is also given the master public key which occurs offline before joining the network. In addition, security mechanisms for wired networks may help in the process of certification.

KPV_M	Private key of Master node
KPU_M	Public key of Master node
KPV_A	Private key of node A
KPU_A	Public key of node A
C_A	Certificate of node A
ID_A	Identity of node A
TS	Timestamp
VP	Validity Period
#	Sequence number of route request
KPV_A <D>	Digital Signature of data D with KPV_A
KPU_A <D>	Encryption of data D with KPU_A
g^x_A	Diffie-Hellman public value of node A
$K_{A,B}$	Diffie-Hellman shared secret key for node A and node B
AC_A	Attribute Certificate of node A
CHA_A	Random Challenge String from node A

Table 4.1 Table of Notation

The certificates are exchanged whenever two nodes interact for the first time. The contents of the certificate are the identity of node ***S***, the public key of node ***S***, and the validity period of the certificate, i.e.

$$C_S = KPV_M <ID_S + KPU_S + VP> \tag{1}$$

4.2.3 Route Discovery Request

A route discovery request is initiated by a source node whenever it has data to send but has no available route. For this purpose, Route Requests (RREQ) are broadcasted by the source and propagated through the network. Since the effectiveness of the network depends on maintaining the correct routing information, a source node must sign the RREQ for integrity before broadcasting it along with its own certificate to its neighbour node. The elements that need to be signed are the identities of source and destination nodes, the sequence number of route request and the Diffie-Hellman public value (i.e. $g\hat{}x_S$) of the source node.

Source node ***S*** begins the route discovery process to destination node ***D*** by broadcasting to its neighbours (e.g. node ***A***) the signed RREQ and its own certificate (Figure 4.1). Source node ***S*** can use different $g\hat{}x_S$ every time it originates a request so that the Diffie-Hellman shared secret key (e.g. $K_{S,D}$=$g\hat{}x_Sx_D$) is different for different session. On the other hand, for the sake of computational efficiency, source node ***S*** may use identical $g\hat{}x_S$ for multiple sessions and let the Diffie-Hellman shared secret key be given by $K_{S,D} = h(g\hat{}x_Sx_D|\#)$ because $g\hat{}x_S$ and $g\hat{}x_D$ need not be computed frequently.

$$S \rightarrow A: \{RREQ, KPV_S<ID_S ,ID_D ,\#,g\hat{}x_S>, C_S\} \tag{2}$$

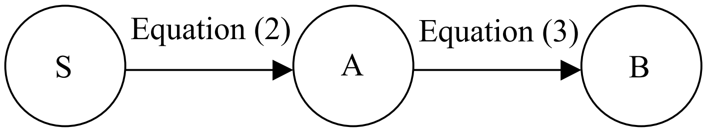
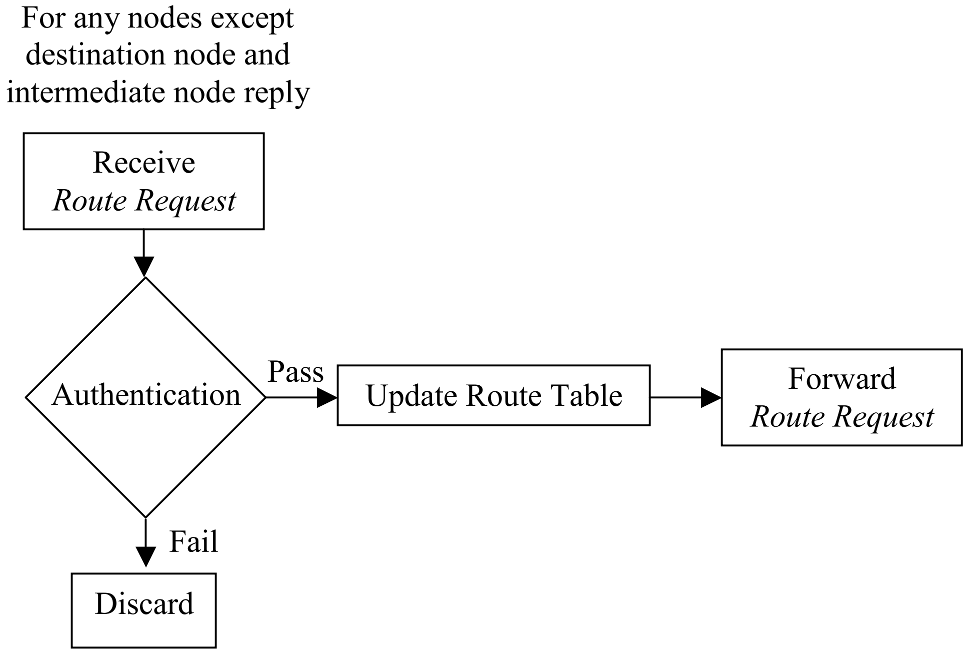


Figure 4.1 Route Discovery Request

The validity of RREQ is verified by using node ***S***’s public key (KPU_S) which can be obtained from the certificate of node ***S***. If RREQ is valid, as shown in Flowchart 4.1, node ***A*** will update its routing table before forwarding the whole route discovery request packet to its neighbour nodes (e.g. node ***B***), as illustrated in Figure 4.1. Otherwise, the route request packet will be discarded.

$$A \rightarrow B: \{RREQ, KPV_S{<}ID_S, ID_D, \#, g\hat{}x_S{>}, C_S\} \tag{3}$$

Node ***B*** will repeat the same step as described above after it has received the request from node ***A***. The route discovery request will be rebroadcast repeatedly until it reaches destination node ***D***. The certificate of node ***S*** may not be useful for node ***A*** because node ***A*** is the immediate neighbour of node ***S***. However, the certificate will become useful after node ***A*** rebroadcasts the request to its neighbour node ***B*** because node ***B*** may not be the immediate neighbour of node ***S***. Thus, node ***B*** may not possess the public key of node ***S*** for this case. Every node along the path to ***D*** needs the certificate of node ***S*** for authentication.

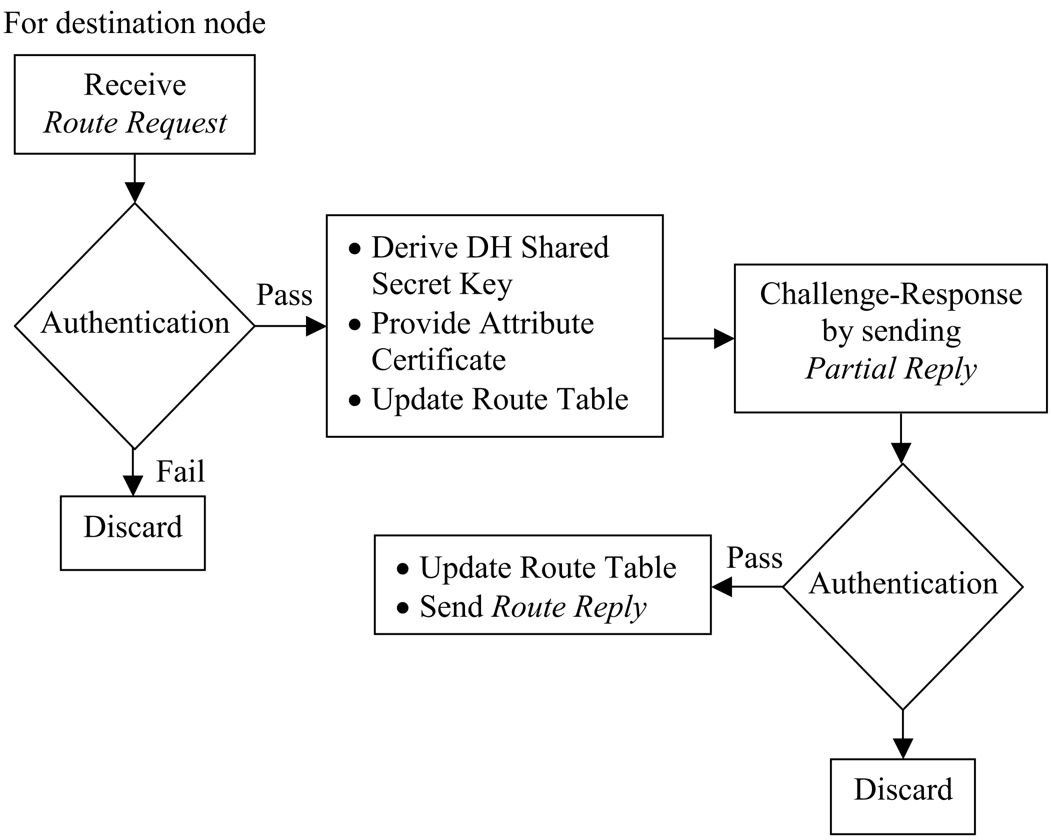


Flowchart 4.1 Route Discovery Request

4.2.4 Route Reply

Destination node ***D*** validates the received route discovery request by confirming the validity of RREQ, as shown in Flowchart 4.2. A Diffie-Hellman shared secret key between the node ***S*** and node ***D*** is derived by node ***D*** (i.e. $K_{S,D}$=g^$x_S x_D$) through the authentication procedure. This shared secret key is used to encrypt data packet in the subsequent transmission. The encryption algorithm is based on symmetric key encryption and this can reduce the computational cost of encryption as compared to public key encryption.

To allow intermediate node reply, which will be discussed in the following section, node ***D*** provides its neighbour an Attribute Certificate (AC_D) whenever it replies to the request. In addition, a node may also provide its immediate neighbours an attribute certificate periodically. The contents of the attribute certificate are the identities of the nodes that are eligible to send a reply on behalf of the attribute



Flowchart 4.2 Route Reply

certificate's originator, the validity period of the attribute certificate, and the Diffie-Hellman public value of the originator, i.e.

$$AC_D = KPV_D{<}ID_{A,B,C,\ldots} + VP + g\hat{\ }x_D{>} \qquad (4)$$

Strong authentication is performed at all adjacent pairs that transmit route replies, as depicted in Figure 4.2. Node ***D*** sends a Partial Reply (PREP) to its neighbour (e.g. node ***C***) before sending the Route Reply (RREP). The PREP has a field that contains a Random Challenge String (CHA_D) for the neighbour. The challenges are small in size to keep the bandwidth overhead low.

$$D \rightarrow C\text{: } \{PREP\} \qquad (5)$$

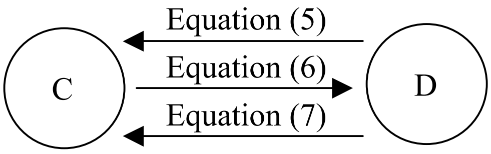


Figure 4.2 Route Reply

Node ***C*** checks the contents of PREP and will discard it if it does not correspond to the current pending route discovery request. Otherwise, node ***C*** will send node ***D*** the encrypted CHA_D and a new challenge string.

$$C \rightarrow D\text{: } \{KPV_C\text{<}CHA_D\text{,}CHA_C\text{>}\} \tag{6}$$

If authentication fails during the challenge-response process, the packet is dropped by node ***D***. Besides sending the random challenge string, node ***D*** sends node ***C*** the entire packet of route reply which consists of the signed RREP, and the attribute certificate.

$$D \rightarrow C\text{: } \{KPV_D\text{<}CHA_C\text{>, } RREP\text{, } KPV_D\text{<}ID_D\text{ ,}ID_S\text{ ,\#,}g\text{\^{}}x_D\text{>, } AC_D\text{, } C_D\} \tag{7}$$

Node ***C*** does the necessary verification. After all these authentications, node ***C*** will keep a copy of node ***D***'s attribute certificate and update its routing table before repeating the same procedure with its neighbour nodes. In addition, node ***C*** needs to send its own attribute certificate to its neighbour as well when relaying the entire route reply packet. Every intermediate node along the path back to the source node ***S*** is required to keep a copy of the relevant attribute certificates as an evidence of having an active route to the corresponding destination node.

The procedure mentioned above continues until the route reply control message reach as the source node ***S***. Node ***S*** updates its routing table after doing the necessary verification and will derive the Diffie-Hellman shared secret key based on the received g^x_D. Since the replies are authenticated, these routes are valid and can be used for sending data packets. Whenever node ***S*** originates a data packet to the intended destination, the data packet must be encrypted with the corresponding secret key for confidentiality.

4.2.5 Intermediate Node Reply

In order to achieve the required robustness and to improve the effectiveness of the route discovery process, an intermediate node ***I*** can generate an Intermediate Node Reply (INREP) provided that it has an active route to the destination and also has the valid attribute certificate chain to the destination. This is the only case where the route discovery request does not actually reach the destination. It is required to present the attribute certificate chain in order to prevent a node from generating a false INREP. Furthermore, an Intermediate node cannot reply to the route discovery request although it has the active route to destination if one of the attribute certificates has expired.

The intermediate node reply procedure is likely to be the same as the destination node route reply procedure. The Intermediate node ***I*** receives a route request from its neighbours (e.g. node ***B***), as shown in Figure 4.3, and it has the active route to the sought destination. Therefore, node ***I*** will send PREP to node ***B*** to reply to the request. Node ***B*** examines the contents of PREP. If it does not correspond to the present awaiting route discovery request, the packet will be dropped. Otherwise, node ***B*** will send node ***I*** a new challenge string and the encrypted CHA_I.

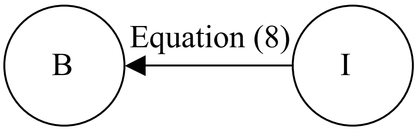


Figure 4.3 Intermediate Node Reply

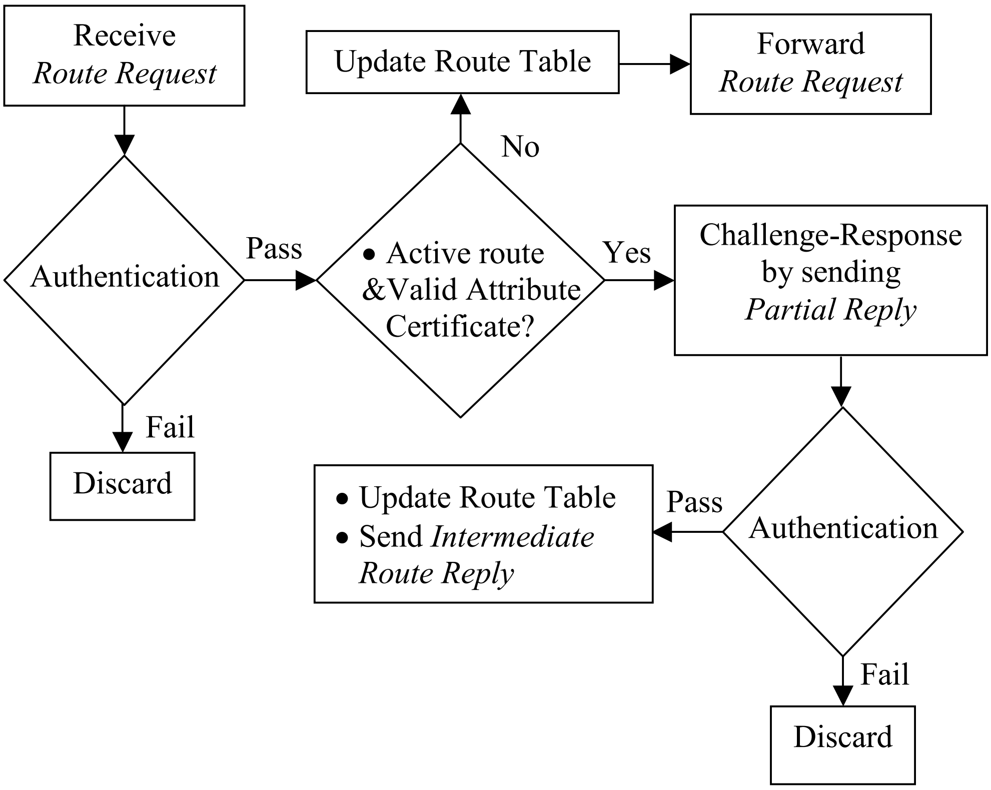
After the necessary authentication from challenge-response identification, node ***I*** will send node ***B*** the whole intermediate node reply packet which comprises of the signed INREP, and the relevant attribute certificate chain, as illustrated in flowchart 4.3.

$$I \rightarrow B\text{: } \{KPV_I\text{<}CHA_B\text{>}, INREP, KPV_I\text{<}ID_I, ID_S, \#\text{>}, (AC_I, \ldots, AC_D), C_D\} \qquad (8)$$

Signing the INREP is to guarantee the non-repudiation of the sender. The packet is discarded if the authentication fails. After the authentication, node ***B*** updates its routing table and repeats the same authentication procedures with its neighbours who sent the identical RREQ to node ***B***. The procedures described above will continue until the control messages arrive at source node ***S***. In this case, the Diffie-Hellman shared secret key between the node ***S*** and node ***D*** can be derived by obtaining $g\text{^}x_D$ from node ***D***'s attribute certificate. In order to maintain the secrecy of the data packet, it must be encrypted with the secret key whenever it is transmitted to the intended destination.

In SEROP, we cannot verify whether node ***I*** still has an active route (link may be broken) to the destination although it possesses the valid attribute certificate chain. However, node ***I*** is unable to deny having sent the control messages in the past due to the non-repudiation property applied to the signed INREP. Furthermore, by setting short validity period for attribute certificate, it can minimize the harm to the network caused by false INREP replies from node ***I***.

For Intermediate Node Reply



Flowchart 4.3 Intermediate Node Reply

4.2.6 Route Maintenance & Deletion

Route Maintenance takes place when a node (e.g. node ***E***) in the active path loses its entire route to the destination. Nevertheless, node ***E*** attempts a local route repair by sending a RREQ with limited propagation range. If this attempt fails, node ***E*** will instantly carry out Route Deletion and send its upstream node (e.g. node ***U***) the Route Error message (RERR), as shown in Figure 4.4. All RERRs along with the timestamp must be signed by the sender.

$$E \rightarrow U: \{RERR, KPV_E<RERR ,TS>\} \qquad (9)$$

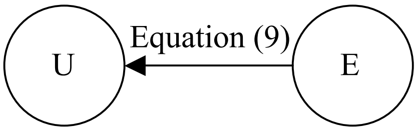
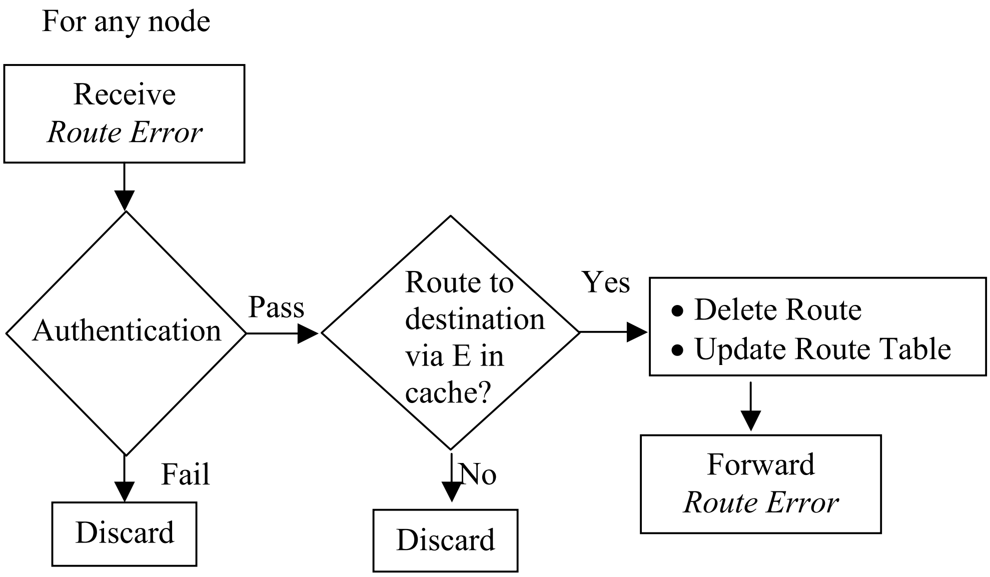


Figure 4.4 Route Maintenance & Deletion

A timestamp ensures that the RERR is fresh. After checking the cache, the corresponding route entry will be deleted by node ***U*** only if ***U***'s route to the deleted destination is via E. Otherwise, RERR will be dropped. Node ***U*** then sends the RERR and repeats the same procedure to its upstream node, as illustrated in Flowchart 4.4.

It is extremely difficult to identify RERRs that are from a link which is not broken. However, the node that generates RERR cannot deny having sent it because the RERR is signed. In addition, there are no advantages at all if a node maliciously generates a RERR to its upstream node since its upstream node may have alternative routes to the destination



Flowchart 4.4 Route Maintenance & Deletion

4.2.7 Key Revocation

The Master node ***M*** can revoke the certificate of a node when the node is no longer having the right to enter the network. For the case when a certificate needs to be revoked, the master node is required to sign the Revoke Certificate (RevokeCert) control message and append it to the RevokeCert before broadcasting it to the ad hoc network.

$$M \rightarrow network: \{RevokeCert,\ KPV_M<RevokeCert>\} \qquad (10)$$

Any node that receives RevokeCert will rebroadcast it to its neighbours and update its routing table to avoid transmitting data through that particular node. The overall picture of the protocol is shown in Figure 4.5.

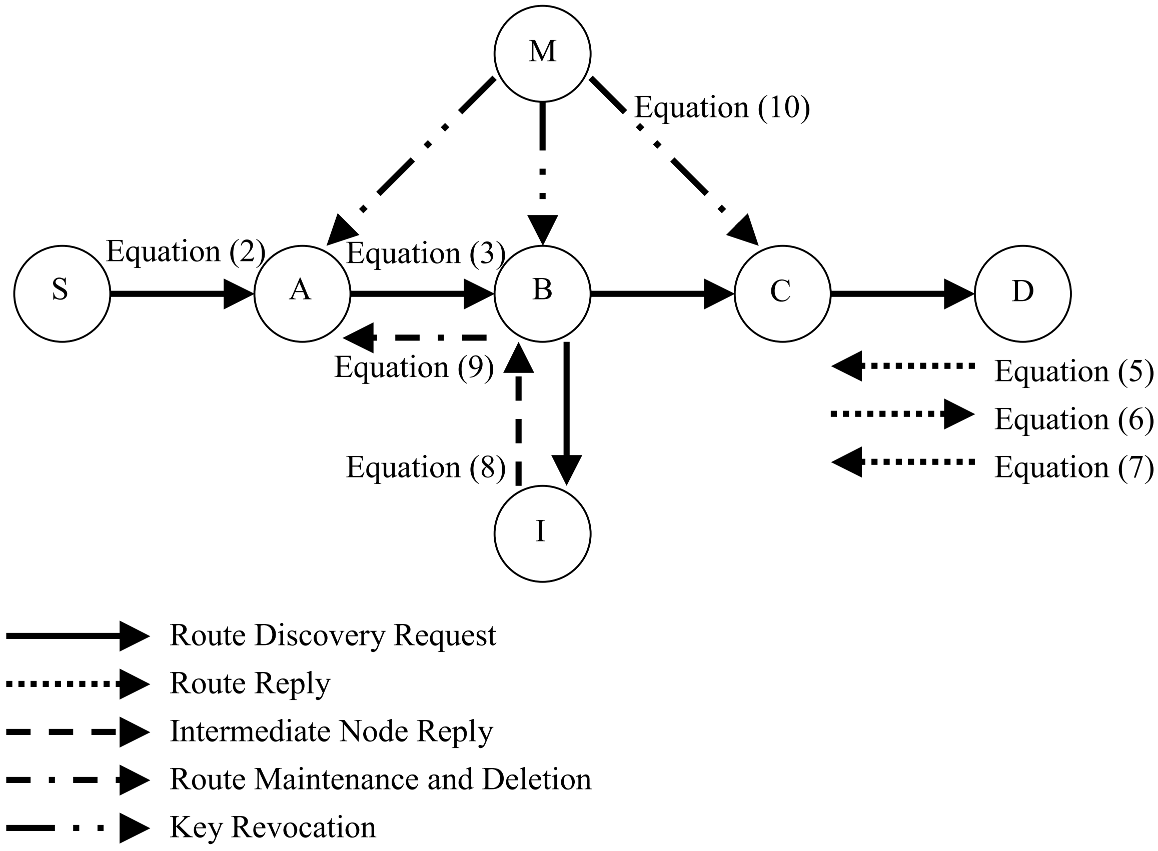
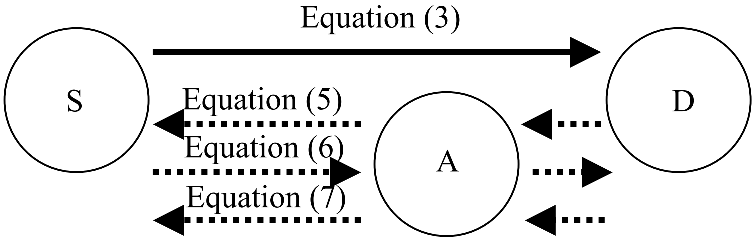


Figure 4.5 Overall Picture of the Protocol

4.3 Reasoning Logic

In this section, we demonstrate the reasoning process to SEROP. The analysis is based on the methodology in [29,28]. Since the protocol is the basis of security in the ad-hoc networks, it is therefore crucial to make sure that the protocol is functioning correctly. Based on certain initial beliefs and possessions, the participating principals may expand their belief sets, as a result of receiving new beliefs. Inference rules make the derivation of new beliefs from current beliefs and from incoming messages possible. The general assumption is that principals do not disclose their secrets.

In particular, we follow the notations, methods and selected logical postulates underlying our reasoning process in [28,22]. The complete set of the notations is included in table 4.2, as in [28,22]. The line that separates two statements indicates that the upper statement implies the lower one. The protocol is to be in deep thought as the exchange of two control messages, a Route Discovery Request and a Route Reply. The protocol with the parts which contain the control messages that do not contribute to the participants' beliefs is omitted, as shown in the figure 4.6. Node ***S*** and node ***D*** are the source node and destination node respectively. Node ***A*** is the immediate neighbour of source node ***S***.



Equation (3) $\{RREQ, KPV_S<ID_S ,ID_D ,\#,g\hat{}x_S>, C_S\}$
Equation (5) $\{PREP\}$
Equation (6) $\{KPV_S<CHA_A,CHA_S>\}$
Equation (7) $\{KPV_A<CHA_S>, RREP, KPV_D<ID_D ,ID_S ,\#,g\hat{}x_D>, AC_D , C_D\}$

Figure 4.6 The protocol is observed as an exchange of two control messages

$P < X$	P is told formula X
$P < *X$	P is told formula X which he did not convey previously in the current run
$P \ni X$	P possesses, or is capable of processing, formula X
$P \mid\sim X$	P once conveyed formula X
$P \mid\equiv \#(X)$	P believes, or is entitled to believe, that formula X is fresh
$P \mid\equiv \phi(X)$	P believes, or is entitled to believe, that formula X is recognizable
$P \mid\equiv \overset{KPU_Q}{\alpha} Q$	P believes, or is entitled to believe, that KPU_Q is a suitable public key for Q
$P \mid\equiv P \xleftrightarrow{K_{P,Q}} Q$	P believes, or is entitled to believe, that $K_{P,Q}$ is a suitable secret for P and Q
$P \mid\equiv Q \mid\Rightarrow C$	P believes that Q has jurisdiction over statement C

Table 4.2 Table of Statement

For *Route Discovery Request*, the initial assumptions are:

(i) $S \ni (KPU_S / KPV_S), S \mid\equiv \overset{KPU_S}{\alpha} S, S \mid\equiv \overset{KPU_M}{\alpha} M$

The source node ***S*** possesses its own public/private key pair and believes that the KPU_S is a suitable public key for itself. It also believes that KPU_M is the suitable public key for the Master node ***M***.

(ii) $D \ni (KPU_D / KPV_D), D \mid\equiv \overset{KPU_D}{\alpha} S, D \mid\equiv \overset{KPU_M}{\alpha} M$

(iii) $A \ni (KPU_A / KPV_A), A \mid\equiv \overset{KPU_A}{\alpha} S, A \mid\equiv \overset{KPU_M}{\alpha} M$

Similarly, destination node ***D*** and node ***A*** hold their own public/private key pairs and believe that the KPU_D and KPU_A is a suitable public key for itself respectively. Both nodes also trust that KPU_M is the right public key for Master node ***M***. Moreover, node ***S*** believes that PREP and RREP are recognizable, and node ***D*** believes that RREQ is recognizable.

(iv) $S \mid\equiv \phi(PREP, RREP), D \mid\equiv \phi(RREQ)$

For *Equation (3),* we have:

(v) $\dfrac{D < *(RREQ, KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle, Cs)}{D < (RREQ, KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle, Cs)}$ *and*

$$\frac{D < (RREQ, KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle, Cs)}{D \ni (RREQ, KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle, Cs)}$$

This means that node ***D*** receives a packet with the not-originate-here property, and node ***D*** is capable to process the contents of the packet.

(vi) $\dfrac{D \ni (RREQ, KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle, Cs)}{D \ni RREQ, D \ni KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle, D \ni Cs}$ *and* $\dfrac{D \ni Cs}{D \ni KPUs}$

Likewise to (vi), we infer that node ***D*** holds the remainder of the control messages. Since node ***D*** holds the certificate of node ***S***, we also deduce that node ***D*** knows the public key of node ***S***. As a result, node ***D*** believes that KPU$_S$ is the suitable public key for node ***S***.

(vii) $D \mid\equiv \overset{KPUs}{\alpha}\ S$

Then from (iii), (iv), (v), (vi), we obtain:

(viii) $\dfrac{D < KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle, \mathrm{D} \ni \mathrm{KPUs}, D \mid\equiv \overset{KPUs}{\alpha}\ S, D \mid\equiv \phi(RREQ)}{D \mid\equiv S \mid\sim \mathrm{RREQ}, \mathrm{D} \mid\equiv \mathrm{S} \mid\sim KPVs\langle ID_S, ID_D, \#, g^\wedge x_S\rangle}$

This means that node ***D*** believes that node ***S*** once conveyed RREQ and the signature of RREQ. This implies the belief that the control message is originated from node ***S***. Until this stage, the stated goal is fulfilled, i.e. the Route Discovery Request is indeed from node ***S*** and not from other nodes.

For *Route Reply*, the initial assumptions are:

(ix) $A \ni CHA_A, A \mid\equiv \#(CHA_A), S \ni CHA_S, S \mid\equiv \#(CHA_S)$

Node ***A*** is possessed of CHA$_A$ and source node ***S*** possesses CHA$_S$ and believes that CHA$_A$ and CHA$_S$ are fresh respectively.

(x) $D \ni K_{S,D}, D \mid\equiv \#(K_{S,D}),\ D \mid\equiv D \xleftrightarrow{K_{S,D}} S$

Node ***D*** possesses the new generated Diffie-Hellman shared secret key K$_{S,D}$ and believes the key is fresh. Node ***D*** believes that K$_{S,D}$ is a suitable secret key for node ***D*** and node ***S***.

(xi) $\dfrac{S \ni \mathrm{C_A}}{\mathrm{S \ni KPU_A}}, S \mid\equiv \overset{KPU_A}{\alpha}\ A, \dfrac{A \ni \mathrm{C_S}}{\mathrm{A \ni KPU_S}}, A \mid\equiv \overset{KPU_S}{\alpha}\ S$

Since the certificates are exchanged wherever two nodes interact for the first time, thus, node ***S*** and node ***A*** believe that KPU$_A$ and KPU$_S$ are the suitable public keys for node ***A*** and node ***S*** respectively.

(xii) $A \mid\equiv \phi(CHA_A), S \mid\equiv \phi(CHA_S), S \mid\equiv \phi(K_{S,D})$

Node ***A*** believes that CHA$_A$ is recognizable, and node ***S*** believes that CHA$_S$ and K$_{S,D}$ are recognizable.

For *Equation (5),* we have:

(xiii) $\dfrac{S < *\mathrm{PREP}}{S < \mathrm{PREP}}, \dfrac{S < \mathrm{PREP}}{S \ni \mathrm{PREP}}, \dfrac{S \ni PREP}{\mathrm{S \ni CHA_A}}$

i.e. node ***S*** received a Partial Reply (PREP) packet and node ***S*** is capable to process it. Besides that, node ***S*** also possesses CHA$_A$.

Similarly for *Equation (6),* we get:

(xiv) $\dfrac{A < *(\mathrm{KPV_S}\langle \mathrm{CHA_A}, \mathrm{CHA_S}\rangle)}{A < (\mathrm{KPV_S}\langle \mathrm{CHA_A}, \mathrm{CHA_S}\rangle)}\ and\ \dfrac{A < (\mathrm{KPV_S}\langle \mathrm{CHA_A}, \mathrm{CHA_S}\rangle)}{A \ni (\mathrm{KPV_S}\langle \mathrm{CHA_A}, \mathrm{CHA_S}\rangle)}$

Node ***A*** is capable to process the content of the packet with the not-originate-here property.

$$\text{(xv)}\ \frac{A < \mathrm{KPV_S}\langle \mathrm{CHA_A}, \mathrm{CHA_S}\rangle, A \ni KPU_S}{\mathrm{A} < \mathrm{CHA_A}}$$

Similarly to (xiii), we infer that node ***A*** possesses the rest of the field of the control messages. Since node ***A*** possesses the public key of node ***S***, thus node ***A*** is considered to have been told the $\mathrm{CHA_A}$.

Then, from (x), (xi), (xiii), we get:

$$\text{(xvi)}\ \frac{A < \mathrm{KPV_S}\langle \mathrm{CHA_A}, \mathrm{CHA_S}\rangle, \mathrm{A} \ni \mathrm{KPU_S}, \mathrm{A} \mid\equiv \overset{KPU_S}{\alpha}\ S, A \mid\equiv \phi(CHA_A)}{A \mid\equiv S \mid\sim CHA_A, \mathrm{A} \mid\equiv \mathrm{S} \mid\sim \mathrm{KPV_S}\langle \mathrm{CHA_A}, \mathrm{CHA_S}\rangle}$$

Node ***A*** believes that the control message is originated from node ***S***. This signifies the belief that the intended recipient for $\mathrm{CHA_A}$ is in fact node ***S***, and not other nodes. Identity of node ***S*** is verified.

In the same manner, for *Equation (7),* we get:

Let R be the *Equation (7)*,

$$\text{(xvii)}\ \frac{S < *R}{S < \mathrm{R}}\ \mathit{and}\ \frac{S < \mathrm{R}}{S \ni R}$$

$$\text{(xviii)}\ \frac{S < \mathrm{KPV_A}\langle \mathrm{CHA_S}\rangle, \mathrm{S} \ni \mathrm{KPU_A}, \mathrm{S} \mid\equiv \overset{KPU_A}{\alpha}\ A, S \mid\equiv \phi(CHA_S)}{S \mid\equiv A \mid\sim CHA_S, \mathrm{S} \mid\equiv \mathrm{A} \mid\sim \mathrm{KPV_A}\langle \mathrm{CHA_S}\rangle}$$

This time the identity of node ***A*** is verified by node ***S***.

$$\text{(xix)}\frac{S \ni \mathrm{C_D}}{\mathrm{S} \ni \mathrm{KPU_D}}, S \mid\equiv \overset{KPU_D}{\alpha}\ D$$

$$\text{(xx)}\ \frac{S < \mathrm{KPV_D}\langle ID_D, ID_S, \#, g^\wedge x_D\rangle, \mathrm{S} \ni \mathrm{KPU_D}, \mathrm{S} \mid\equiv \overset{KPU_D}{\alpha}\ D, S \mid\equiv \phi(RREP)}{S \mid\equiv D \mid\sim RREP, \mathrm{S} \mid\equiv \mathrm{D} \mid\sim \mathrm{KPV_D}\langle ID_D, ID_S, \#, g^\wedge x_D\rangle}$$

Node ***S*** believes that RREP and the signature of RREP are originated from node ***D***.

$$\text{(xxi)}\ \frac{S < KPV_D\langle ID_D, ID_S, \#, g^\wedge x_D\rangle, S \ni KPU_D}{\mathrm{S} < g^\wedge x_D}$$

Node ***S*** may then generates the Diffie-Hellman shared secret key, $K_{S,D}$ from the received $g \hat{} x_D$. This key should match with each other. Finally,

$$\text{(xxii)}\ \frac{S \mid\equiv D \mid\Rightarrow S \xleftrightarrow{K_{S,D}} D,\ S \mid\equiv D \mid\equiv S \xleftrightarrow{K_{S,D}} D}{S \mid\equiv S \xleftrightarrow{K_{S,D}} D}$$

Node ***S*** believes that node ***D*** has jurisdiction over $S \xleftrightarrow{K_{S,D}} D$. Node ***D*** believes that $K_{S,D}$ is a suitable secret key shared between node ***S*** and node ***D***, meanwhile node ***S*** believes that $K_{S,D}$ is a suitable secret key share between node ***S*** and node ***D***.

After all these complex inferences, source node ***S*** believes that the entire route reply is originated from destination node ***D***. The reasoning process based on the postulates above leads us to the conclusion that the security goals for the protocol are achieved. In a very similar way, this conclusion can be reached for the case where the Intermediate Node generates the route reply.

4.4 Appraisal of SEROP

In this section, we evaluate the ability of SEROP in meeting fundamental security needs such as confidentiality, authentication, integrity and non-repudiation.

Generation of false control messages such as route request and route reply by non-legitimate nodes: This is infeasible because a non-legitimate node does not possess valid certificates from the master node. Since all the nodes are required to exchange their certificate when they interact for the first time, any control messages from a non-legitimate node will be discarded by legitimate nodes if the initial authentication failed.

A maliciously legitimate node impersonating another node and sending route requests to cause inconsistencies in route table: The immediate neighbour that receives the request can prevent this kind of attack by checking the contents of route requests since senders are required to sign the route request before broadcasting it to its neighbours.

Refusal to forward route requests by a malicious node upon receiving it from its neighbours: This kind of attack is hard to counter; however, the regulated flooding of route requests provides the required robustness. Furthermore, to ensure that the protocol is robust over various topologies, discovery of multiple routes to a particular destination is encouraged.

Tampering of control messages by malicious nodes: This action may produce incorrect route information to the network. The routing operations will be affected if the integrity of data is not guaranteed. Nevertheless, this is impossible to happen in the protocol because the integrity of each control message is ensured by the digital signature for that particular control message.

Creation of false Intermediate Node Reply and Route Error intentionally by malevolent node: This type of attack is extremely difficult to prevent in the network. However, setting short expiration time for attribute certificate can mitigate the impairment caused by the false INREP replies.

Release of message contents: This is a kind of passive attack which is very difficult to detect because the opponents do not invoke any alteration of the data. Nevertheless, SEROP enables legitimate nodes to prevent the opponent from learning the contents of the data packets. It is because at any time when a source node originates a data packet to the sought destination, the data packet must be encrypted with the corresponding secret key for confidentiality.

Chapter 5

Simulations and Results

This section presents the performance evaluation of the Secure Routing Protocol (SEROP) [10], a secure routing protocol for mobile ad hoc networks that helps in achieving data confidentiality as well as securing the routing protocol.

5.1 Introduction

Our goal is to evaluate and determine the impact on various performance metrics of the SEROP protocol in both benign and malicious networks. SEROP is an extension built over a base routing protocol, in this case, CHAMP, by modifying the basic operation code of CHAMP. For a benign network, we compare the performance of SEROP with that of the basic CHAMP protocol (which does not take security into consideration). Besides that, we conducted two different attack scenarios for malicious network to determine its robustness in the presence of malicious nodes. The simulation results show that SEROP performs very well in both network environments.

5.2 Performance Evaluation

The objective of this performance study is to evaluate and determine the impact on various performance metrics and operation of the SEROP protocol in both benign and malicious networks through simulation. Since we based our protocol on the basic operation of CHAMP [1], the existing code of CHAMP is modified to make it secure. For benign networks, we compare the performance of SEROP with that of basic CHAMP (which does not specify any special security measures). Besides that, we also performed simulations to compare the performance of SEROP and CHAMP for different source loads.

For malicious networks, we present a security analysis of SEROP by determining its robustness in the presence of different attack scenarios and varying number of malicious nodes. The *first* attack scenario is where the malicious nodes purposely tamper with the RREQ messages and rebroadcast them to their neighbours. This action may produce incorrect route information to the network. The *second* attack scenario is that of a malicious node impersonating another node by rebroadcasting the modified RREP or INREP to cause inconsistencies in the route table.

We used the following metrics to evaluate the performance of our protocol.

(1) *Packet Delivery Ratio*: The total number of packets received divided by the total number of packets originated.

(2) *End-to-end Delay*: The delay of all packets successfully delivered.

(3) *Normalized Routing Overhead*: The total number of the routing messages originated and forwarded divided by the total number of the data packets received. For malicious network, we use byte overhead instead of normalized routing overhead.

(4) *Byte Overhead*: The amount of the overhead bytes transmitted.

5.3 Simulation Setup

The simulation is implemented on the ns-2 simulator with mobility extensions. The network consists of 50 nodes in a rectangular area of 1500 m x 300 m. The two-ray ground model is used as the radio propagation model. For Medium Access Control protocol, the IEEE 802.11 Distributed Coordination Function is used. Traffic sources used are Constant Bit Rate (CBR) where the rate of the packet generation is four data packets per second. Each data packet is 512 bytes long.

For traffic file, source and destination pairs are randomly generated and spread over the entire network. The scenario file, which decides the mobility of the nodes, is generated using the scene generator of the simulator. The mobility model chosen is the Random Waypoint Model. Each node starts moving from a random start point to a random destination with a speed uniformly chosen between zero and a maximum speed. Once the node reaches the destination, it waits for a pause time before moving towards another randomly selected destination. The mobility of the nodes is dependant on the different pause times. Each simulation is carried out for 120 seconds. Table 5.1 summarizes the parameters used in our simulation.

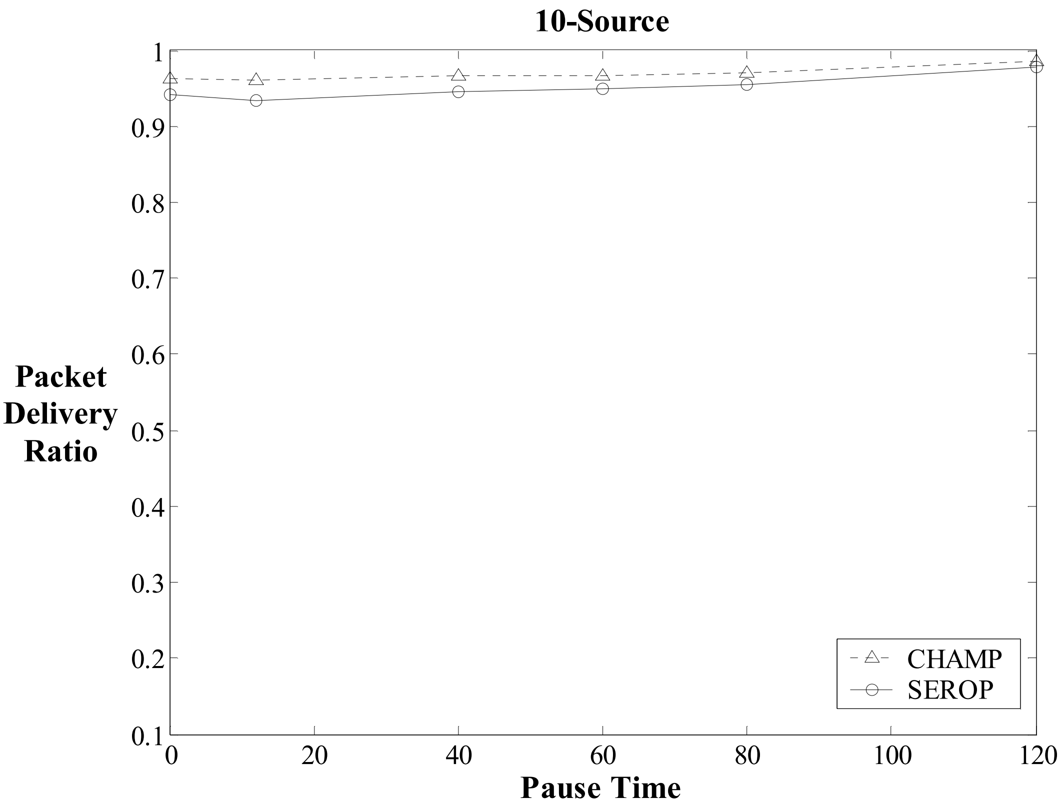
To evaluate the performance of the SEROP protocol as an extension to a reactive routing protocol, we have modified CHAMP in two aspects. The first is increasing the packet size due to the additional overhead of executing the security procedures such as authentication of the control messages. The other is adding another control message to CHAMP i.e. INREP, which enables an intermediate node to reply to the request as stated in the previous section.

Table 5.1 Simulation Parameters

Tool	ns-2 simulator
Number of Nodes	50
Movement Area	1500 m x 300 m
Simulation Time	120 s
Nominal Radio Range	250 m
Raw Bit Rate	2 Mbps
Data Packet Size	512 bytes
Interface Queue	50-packet drop-tail priority
Traffic Type	CBR
Data Rate	4 packets/second
MAC Protocol	IEEE 802.11 DCF
Maximum Speed	20 m/s
Data Collections	60 random runs for benign network 30 random runs for malicious network

5.4 Simulation Results for Benign Network

Figures in this section show the simulation results which compare the performance of SEROP with CHAMP by using the identical traffic source and scenario file. Figure 5.1 shows the Packet Delivery Ratio (PDR) as a function of pause time. Generally, the PDR for all load settings increases with decreasing mobility. At zero pause time for 10 sources, CHAMP delivers 96.3% of packets successfully, and adding security features to CHAMP, SEROP reduces PDR by merely 2%. Specifically, SEROP does not reduce PDR of CHAMP by more than 2.7% for all pause time for 10 sources. At 40 sources, the PDR of SEROP and CHAMP are becoming identical as mobility decreases. This suggests that SEROP performs very well in terms of PDR at low mobility and high traffic load environments.



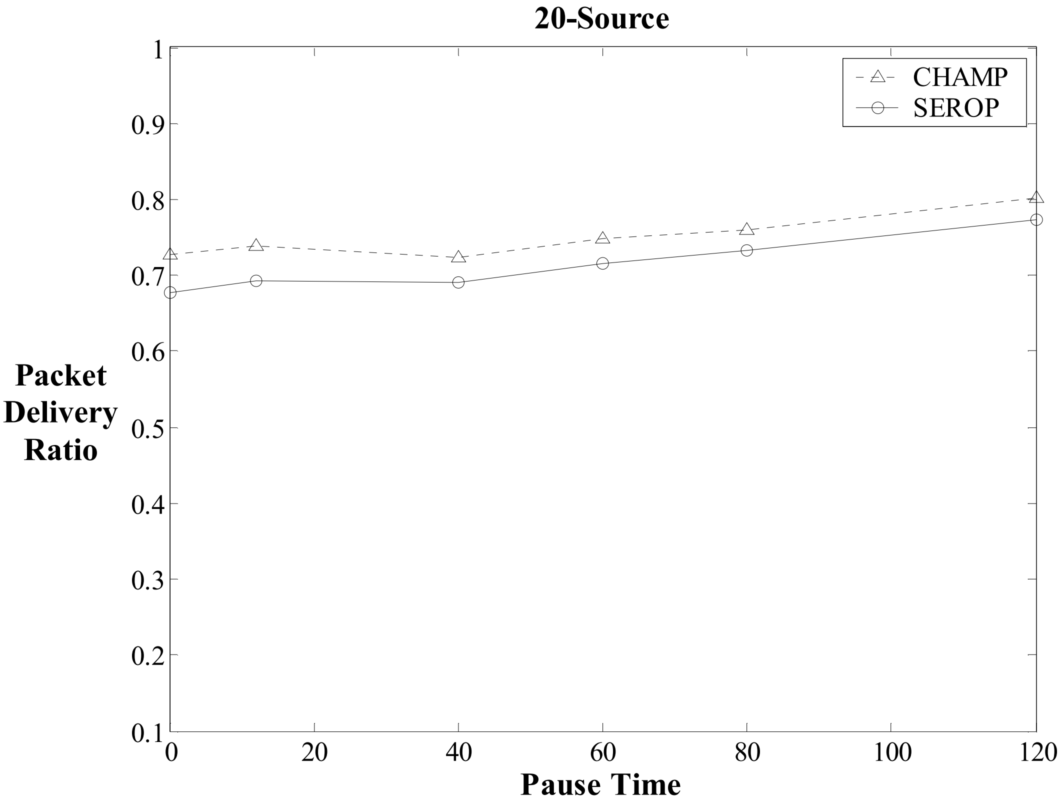
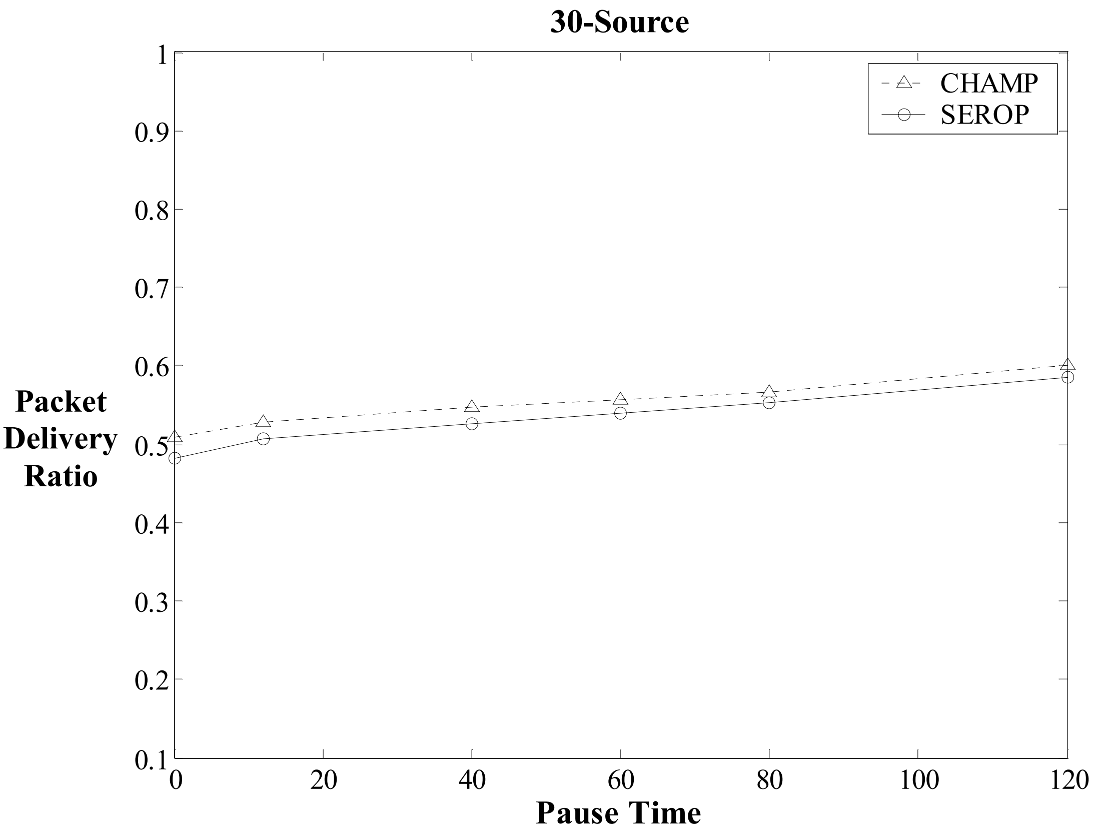


Figure 5.1a Packet Delivery Ratio vs Pause Time



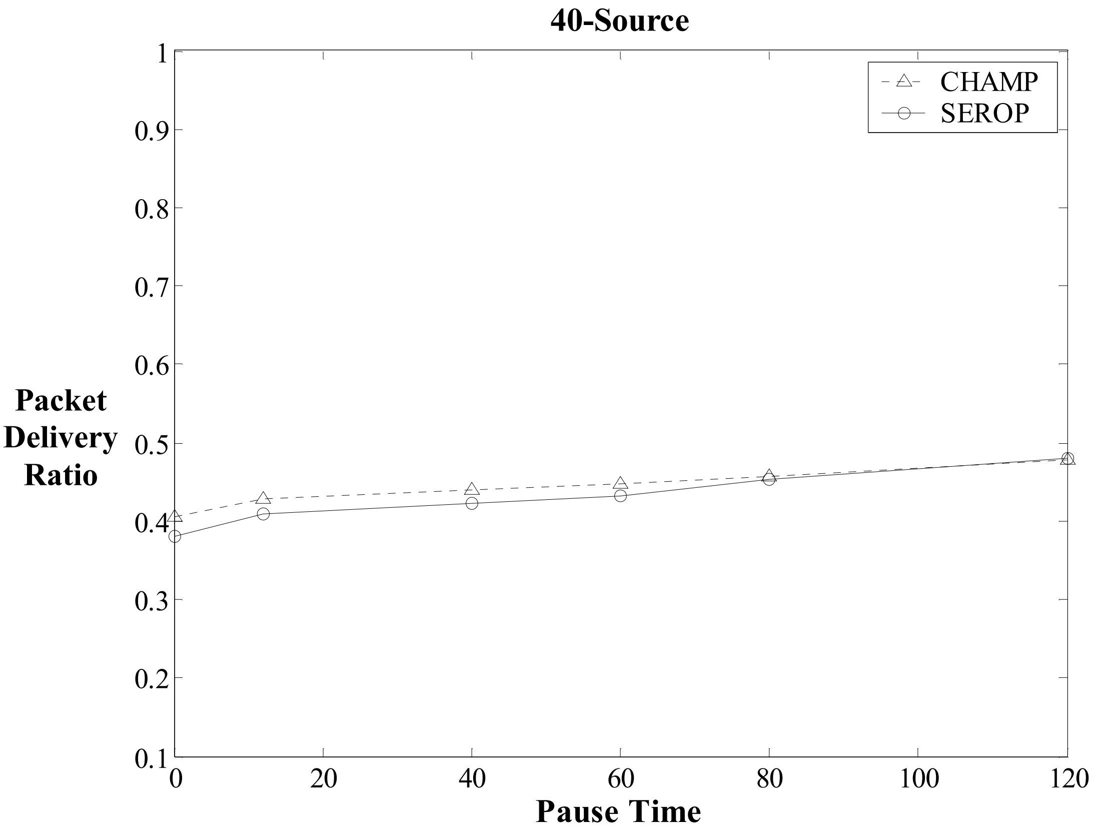


Figure 5.1b Packet Delivery Ratio vs Pause Time

Figure 5.2 illustrates the Packet Delivery Ratio as a function of traffic load. The result on packet delivery is inversely proportional the increase in number of data sources, as depicted in the figure. Basically, the PDR drops as the traffic load increases for all pause time. However, SEROP copes very well with the increase of data sources, as it reduces the PDR by not more than 4% for all number of data sources at all pause time. The results imply that incorporating SEROP into CHAMP does not drastically degrade the performance of the routing protocol in terms of packet delivery.

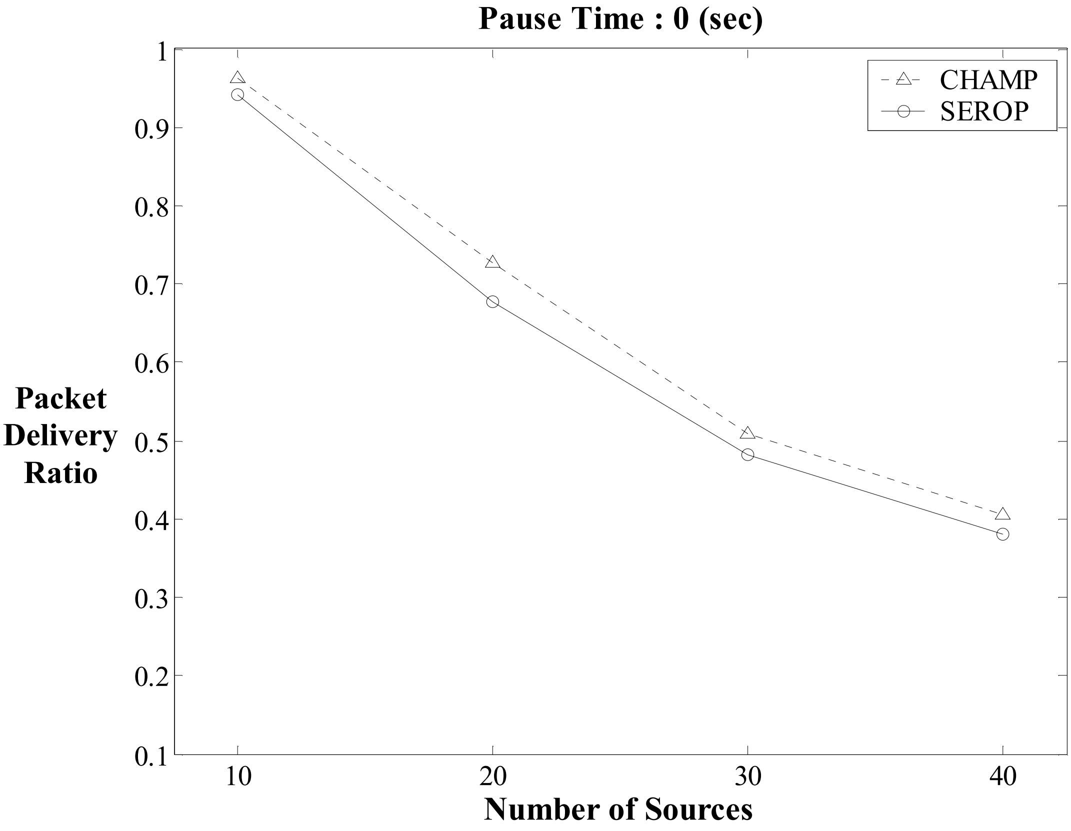


Figure 5.2a Packet Delivery Ratio vs Number of Sources

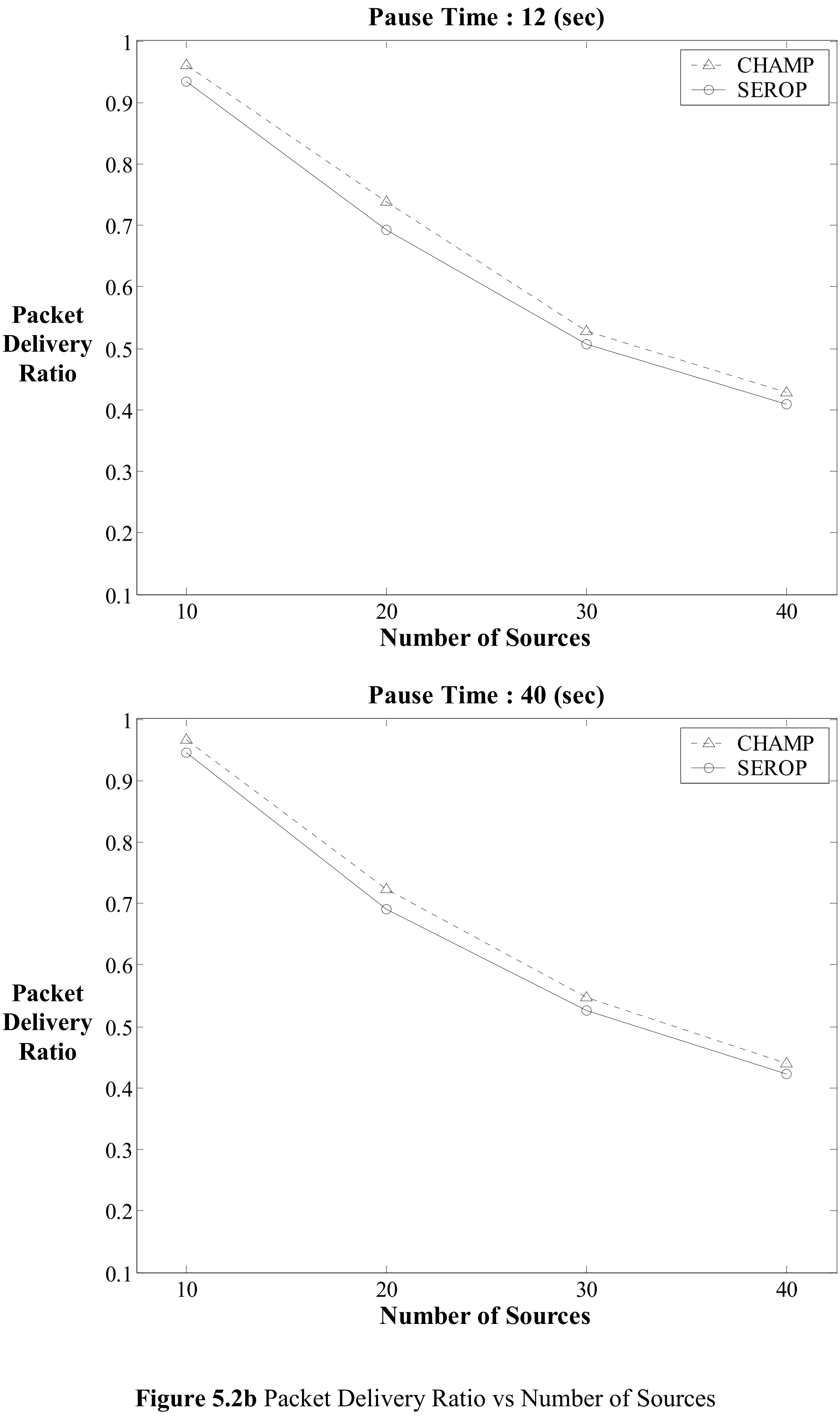
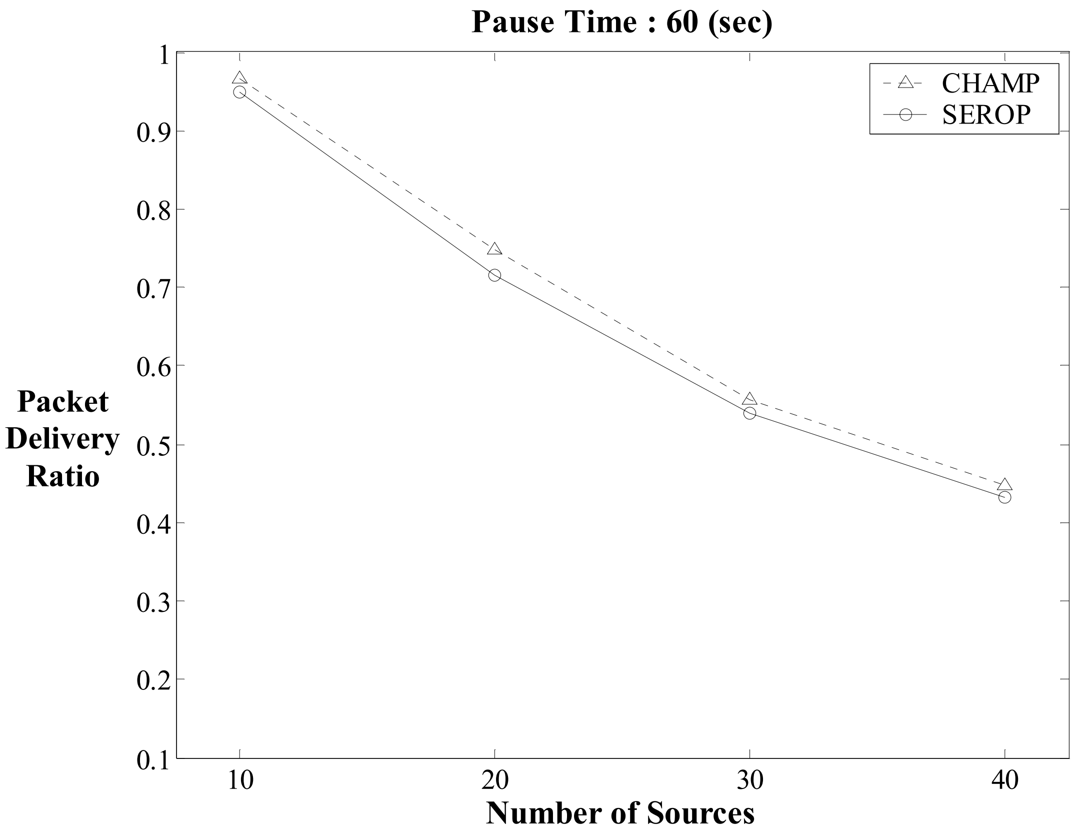


Figure 5.2b Packet Delivery Ratio vs Number of Sources



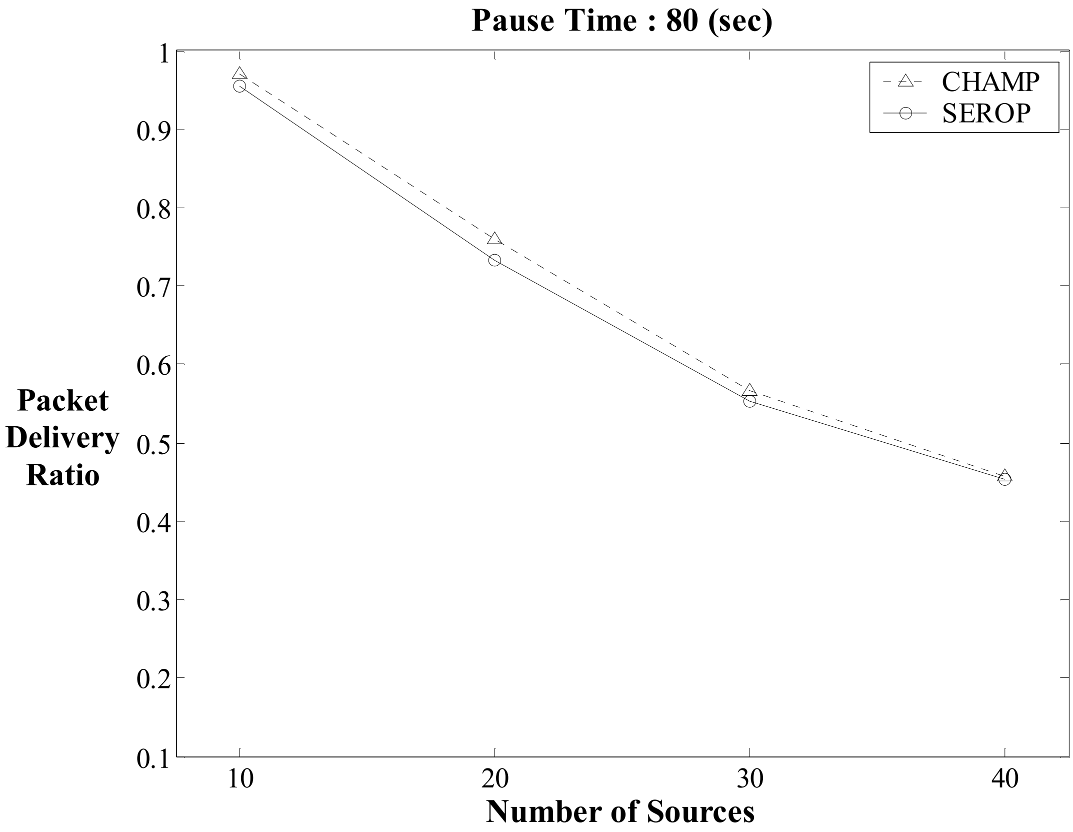


Figure 5.2c Packet Delivery Ratio vs Number of Sources

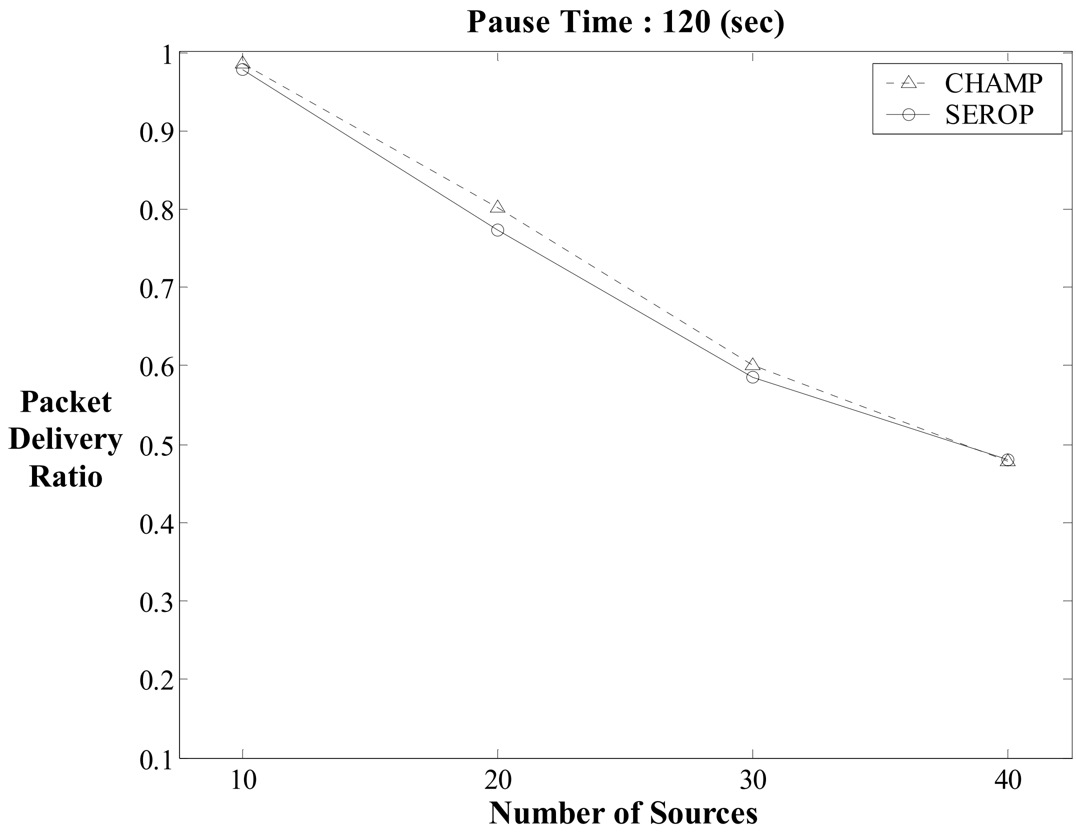


Figure 5.2d Packet Delivery Ratio vs Number of Sources

Figure 5.3 illustrates the average end-to-end delay experienced by the packets. In general, the end-to-end delay for all traffic load increases with decreasing mobility. High mobility rate influences the delay for all protocols. The performance of SEROP in terms of delay is not exception. It produces a low delay at any pause time if compared to CHAMP although required to execute authentication procedures in order to secure the routing protocol. The lowest delay for all pause time is observed when there are only 10 sources.

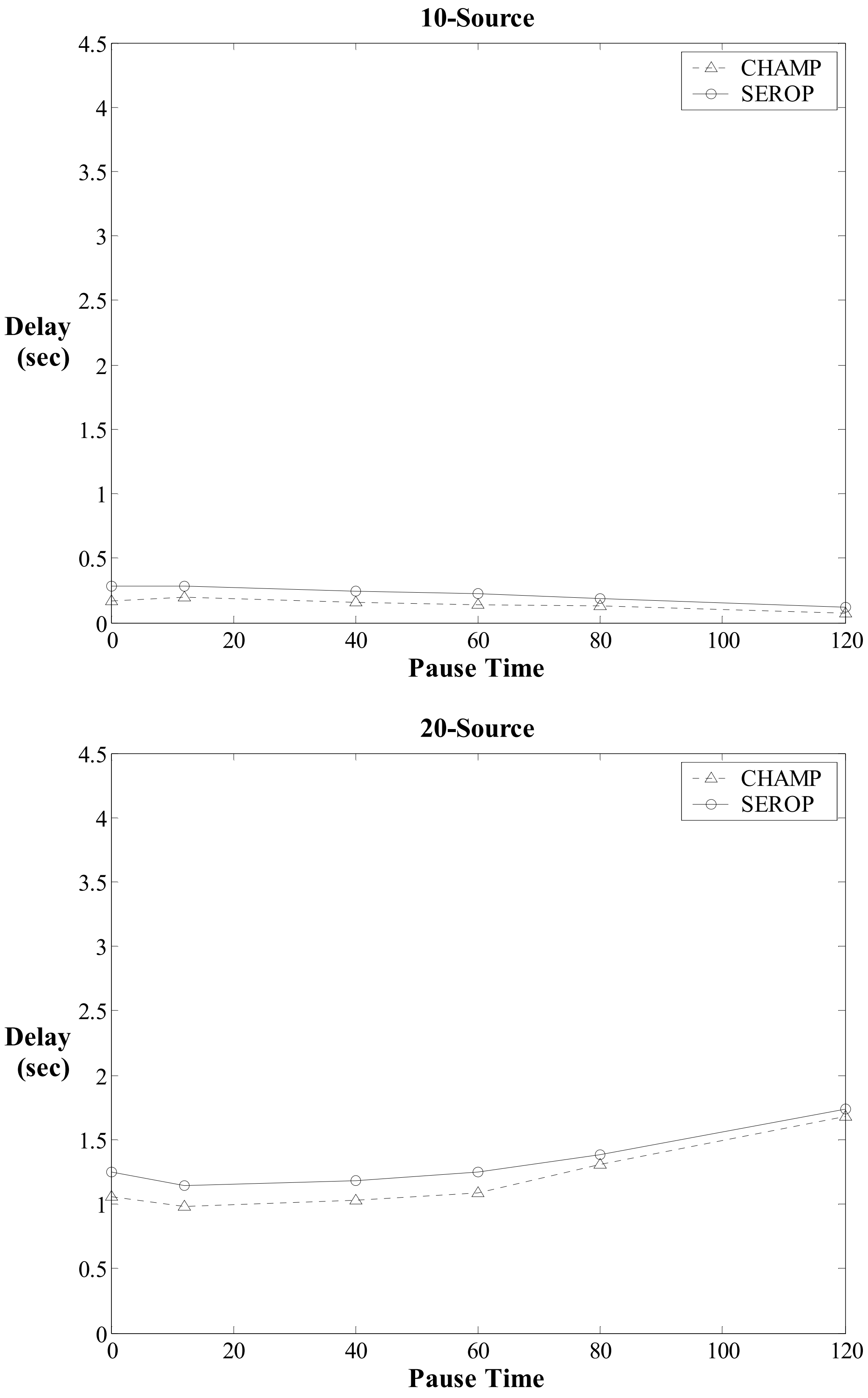
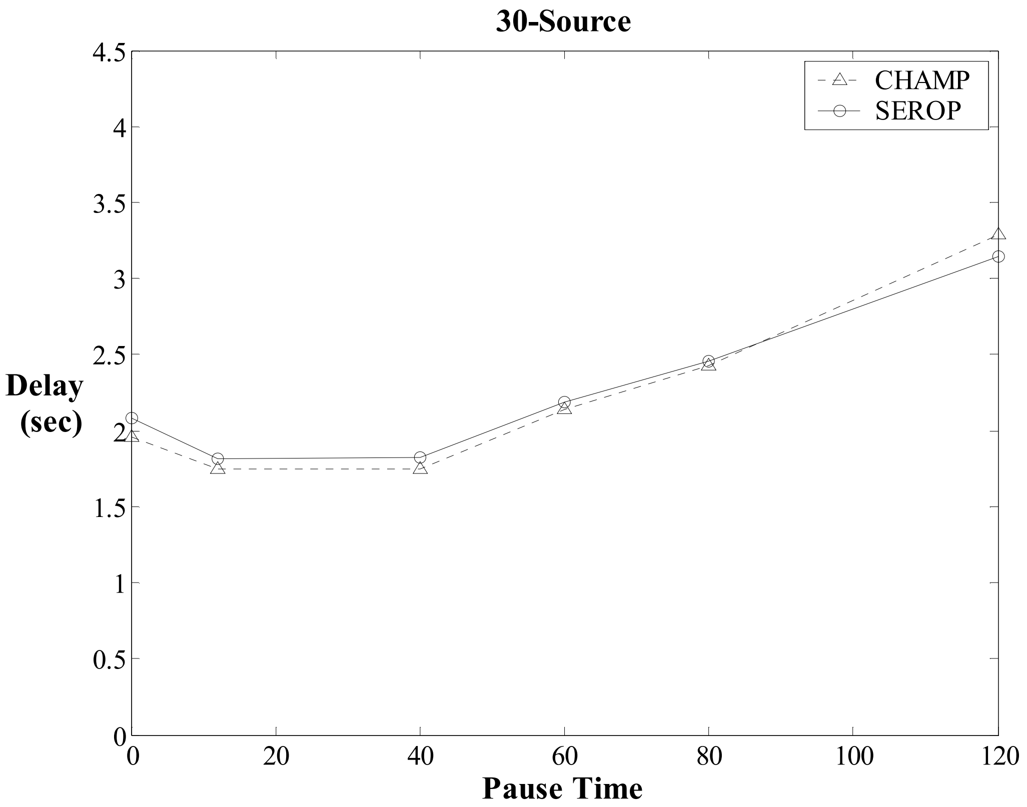


Figure 5.3a End-to-end Delay vs Pause Time



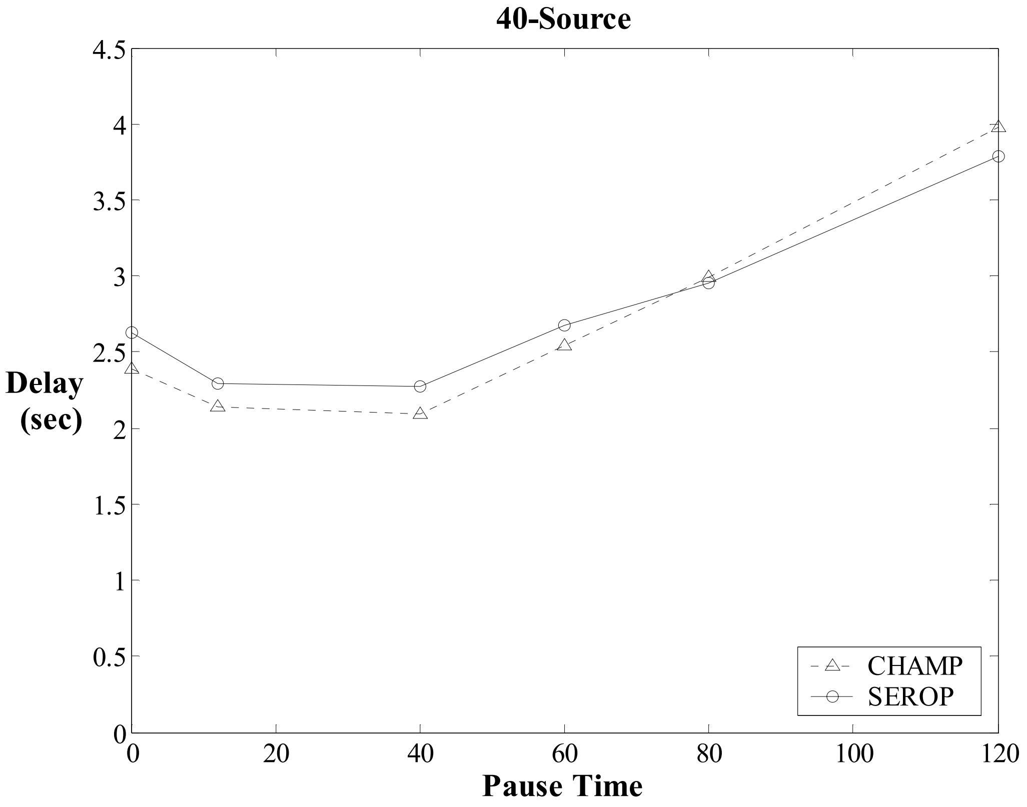


Figure 5.3b End-to-end Delay vs Pause Time

Figure 5.4 shows the average end-to-end delay as a function of different data sources. The delays of the protocols are also influenced by traffic load. For pause times of 0, 12, 40, 60 and 80 seconds, SEROP exhibits a marginal increase of delay over CHAMP. Surprisingly, SEROP outperforms CHAMP at higher pause time, i.e. 120 seconds. This improvement is due to the introduction of intermediate node reply in the protocol. Each node that satisfies the conditions of INREP may help to reduce the delay experienced by the packet by responding to the route query.

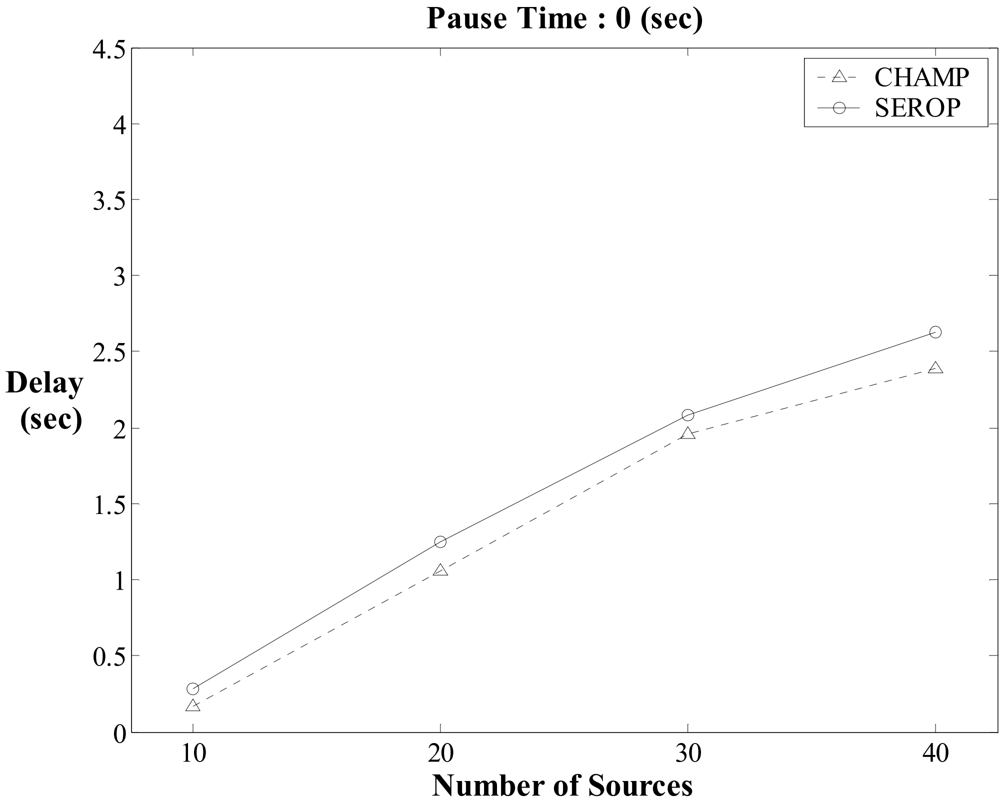
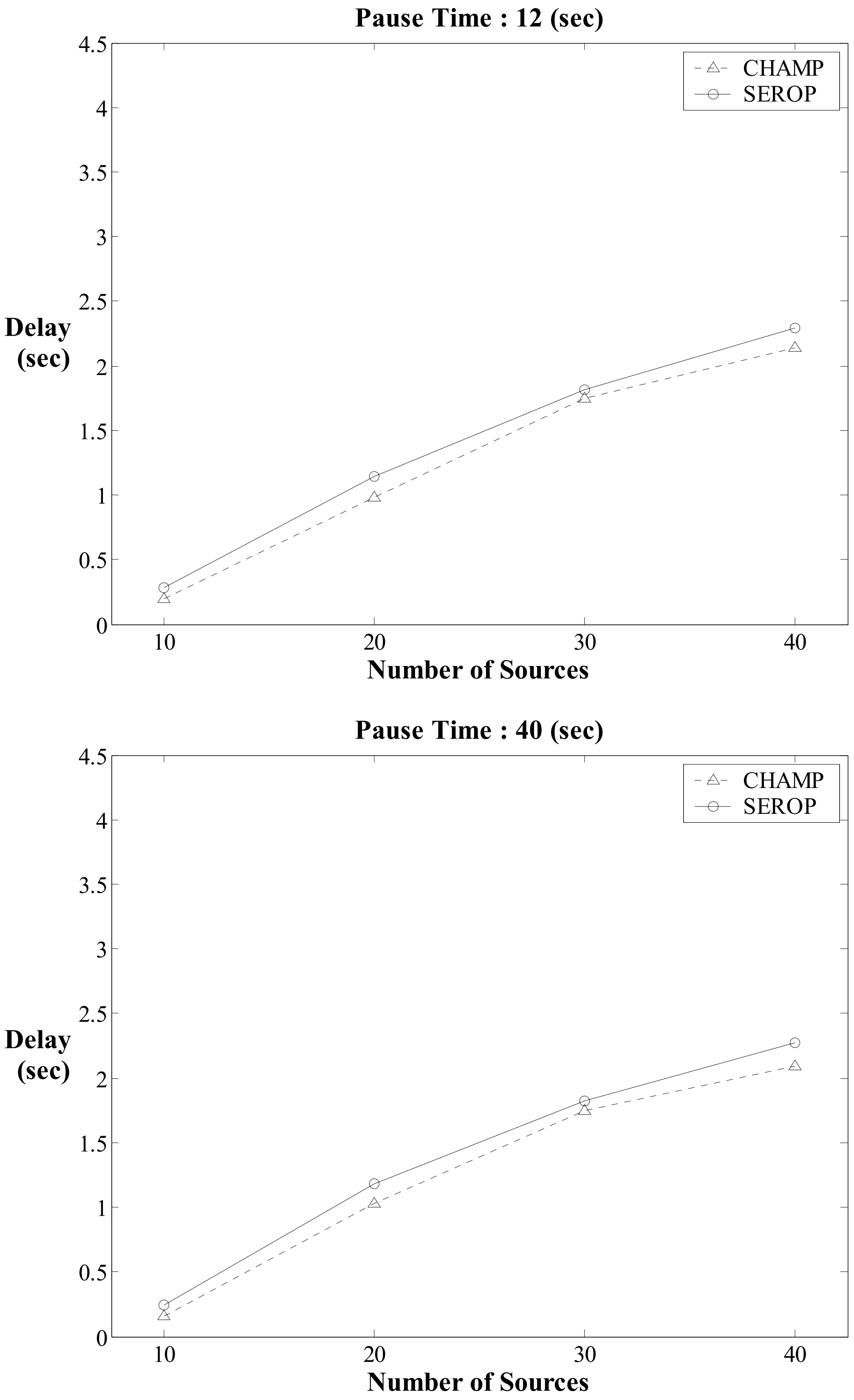


Figure 5.4a End-to-end Delay vs Number of Sources



Pause Time : 12 (sec)

Pause Time : 40 (sec)

Figure 5.4b End-to-end Delay vs Number of Sources

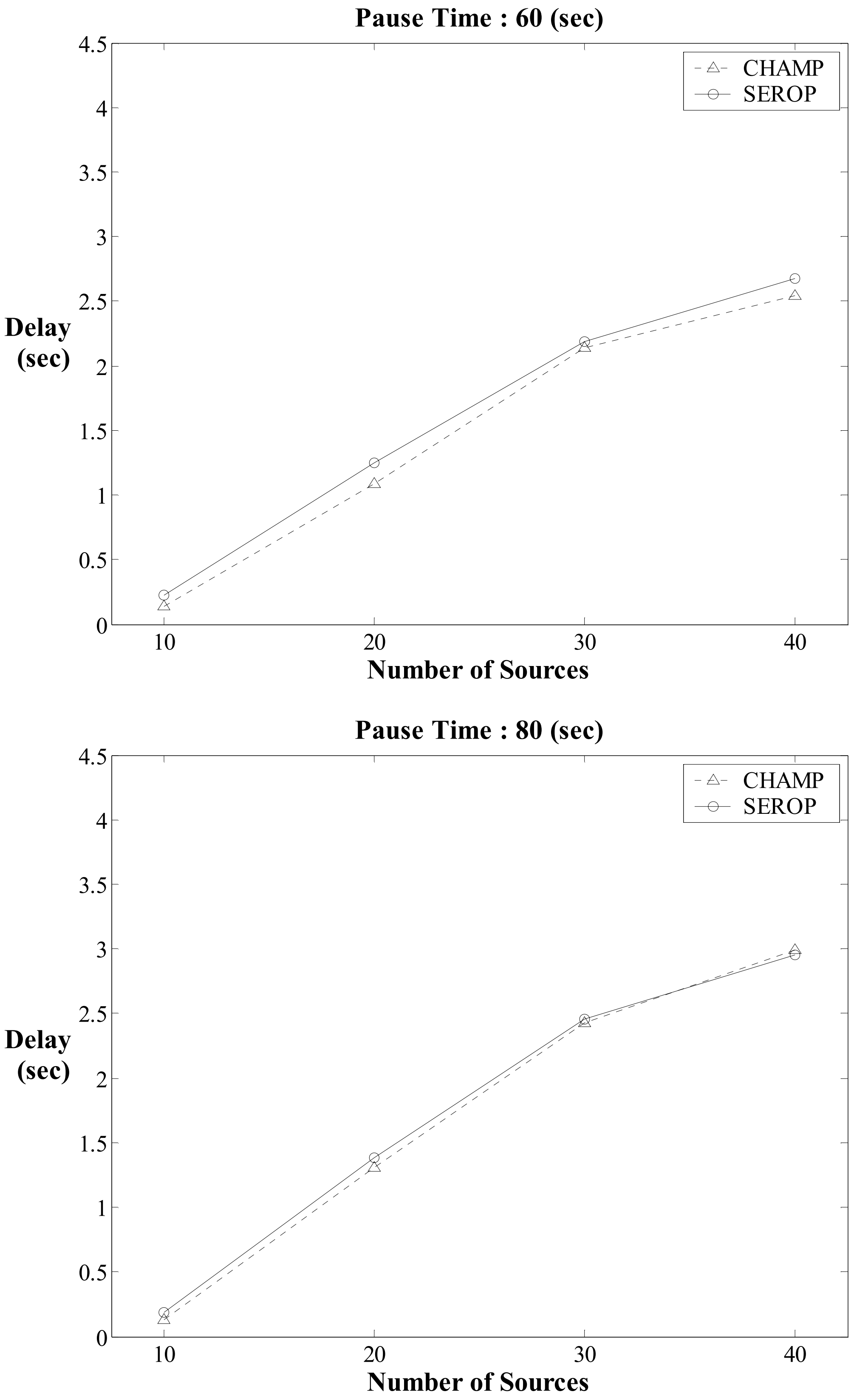


Figure 5.4c End-to-end Delay vs Number of Sources

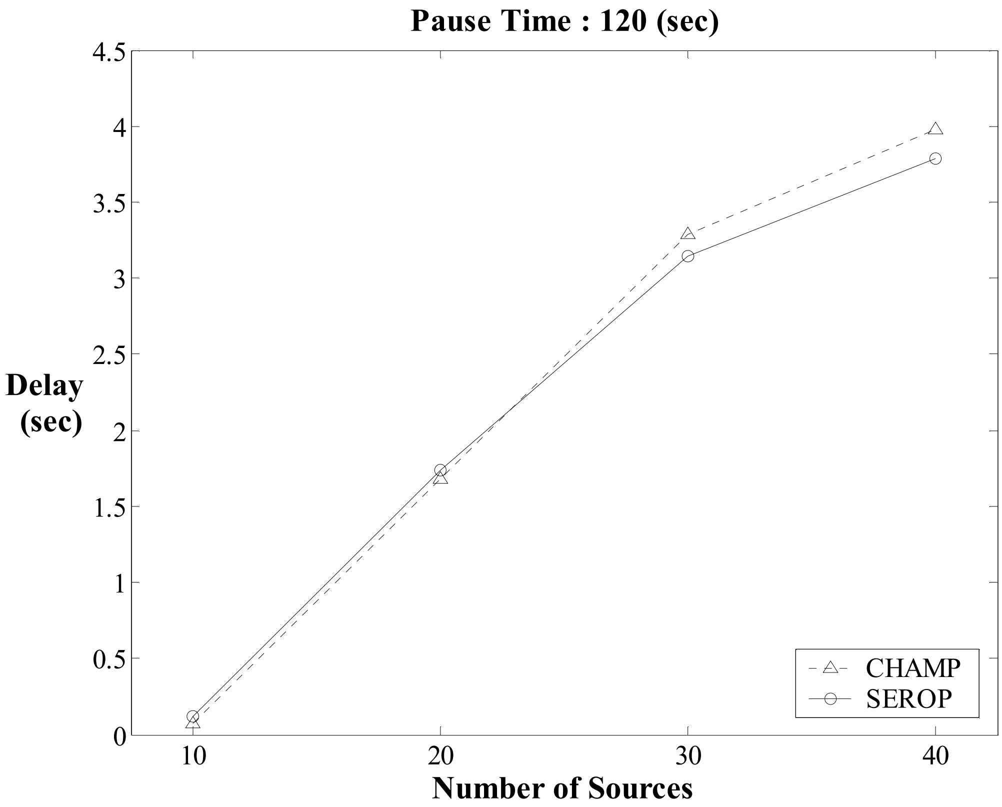


Figure 5.4d End-to-end Delay vs Number of Sources

Since SEROP is a hybrid cryptosystem, it incurs higher overhead than other pure symmetric key security schemes. Figure 5.5 shows the Normalized Routing Overhead as a function of pause time. The increase of mobility also increases the amount of routing overhead. The number of routing packets generated by SEROP and CHAMP increases as the number of data sources increases. Generally, SEROP produces a moderate increase of normalized routing overhead as shown in figure 5.6. However, this is usually the price to pay for the preferred security levels and is an affordable routing overhead.

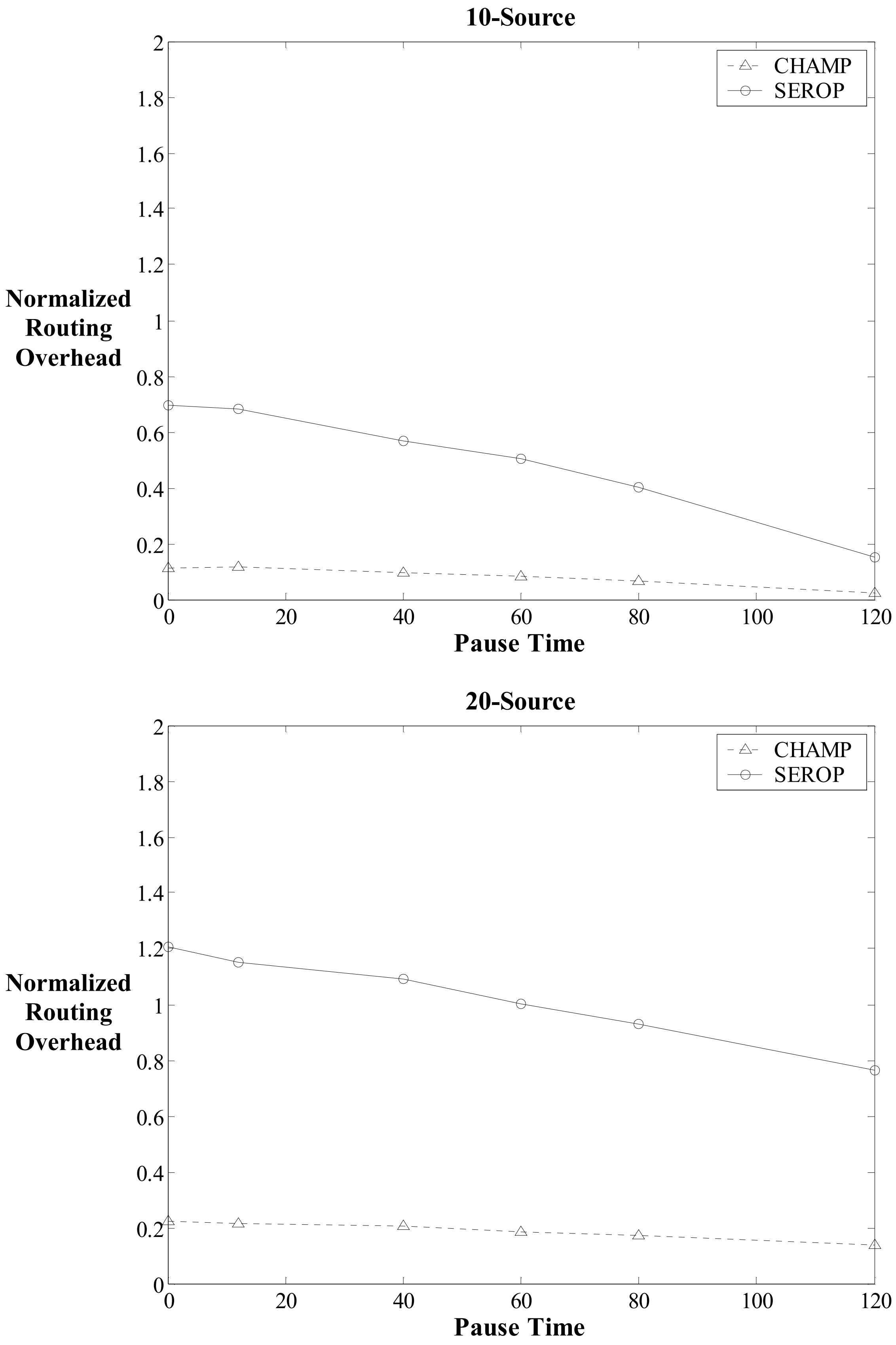


Figure 5.5a Normalized Routing Overhead vs Pause Time

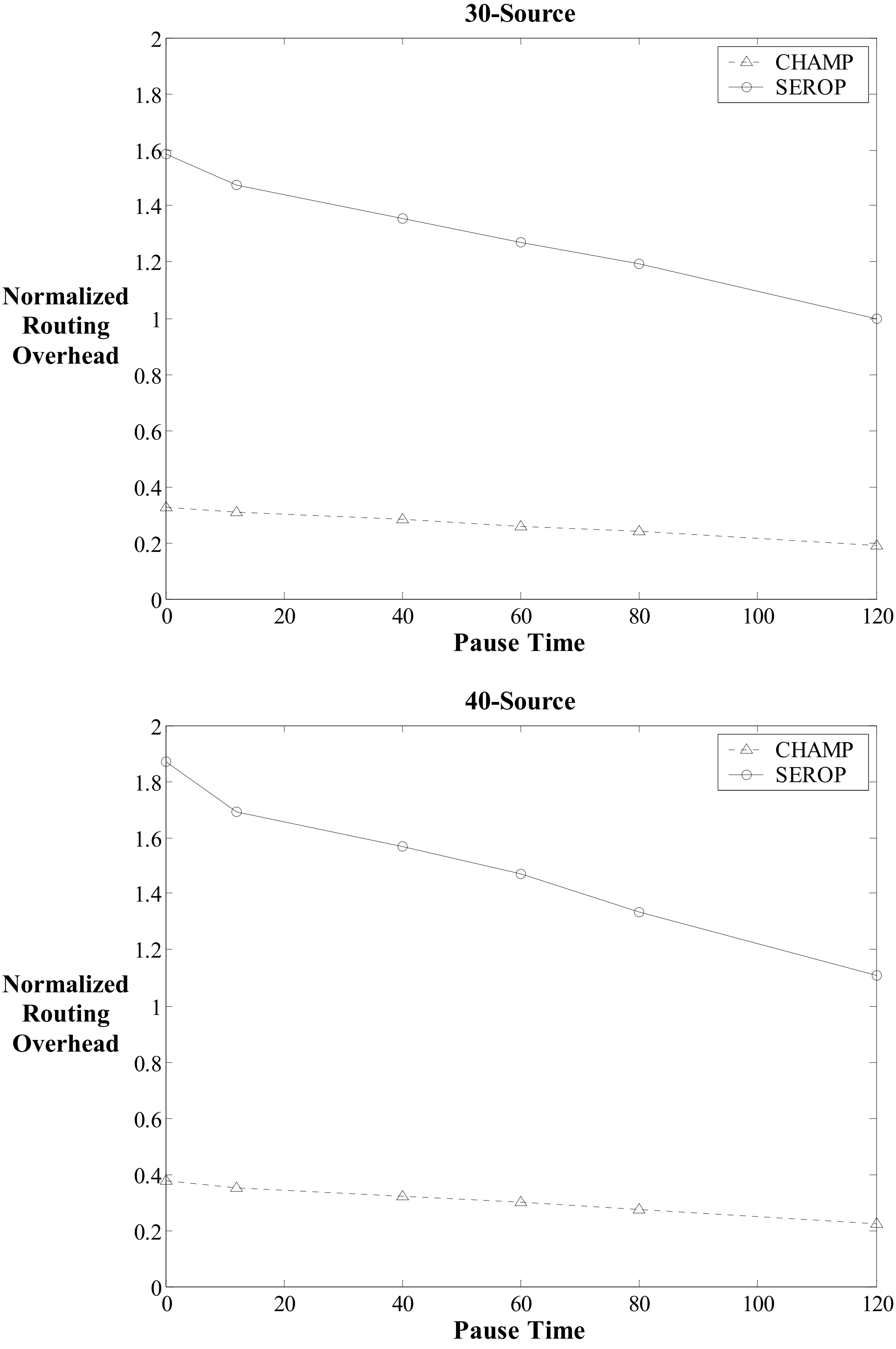
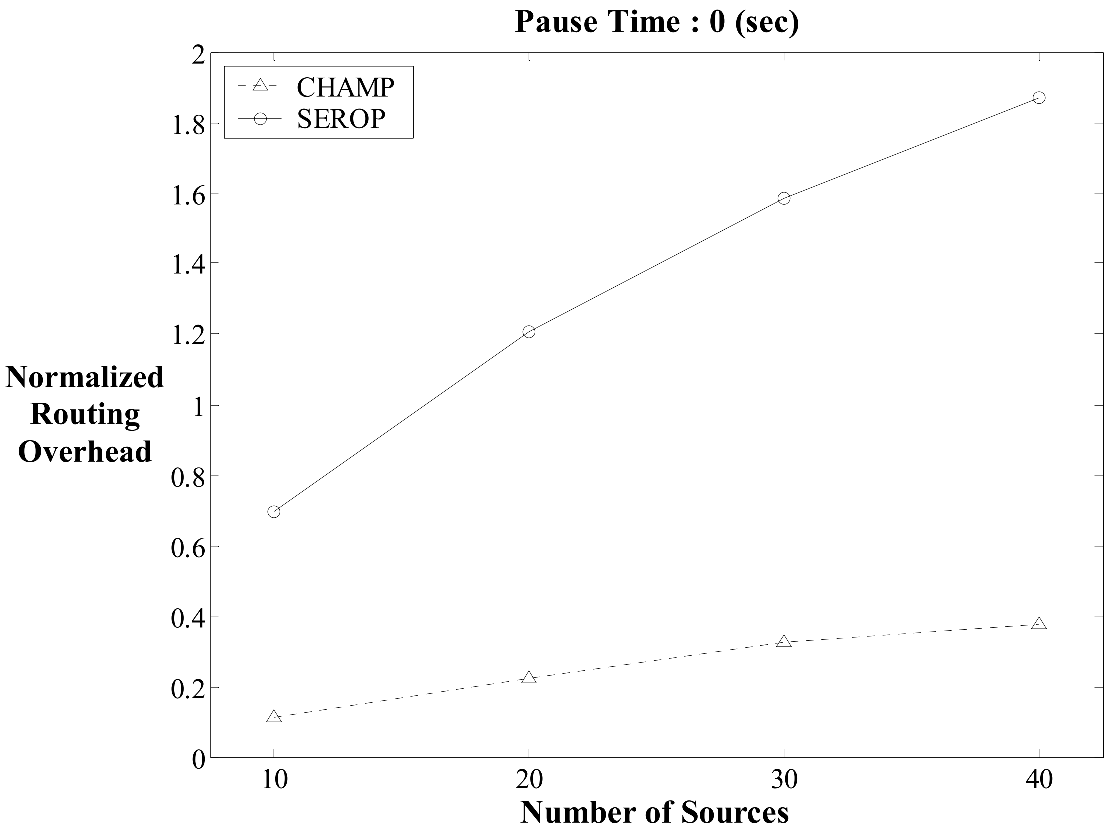


Figure 5.5b Normalized Routing Overhead vs Pause Time



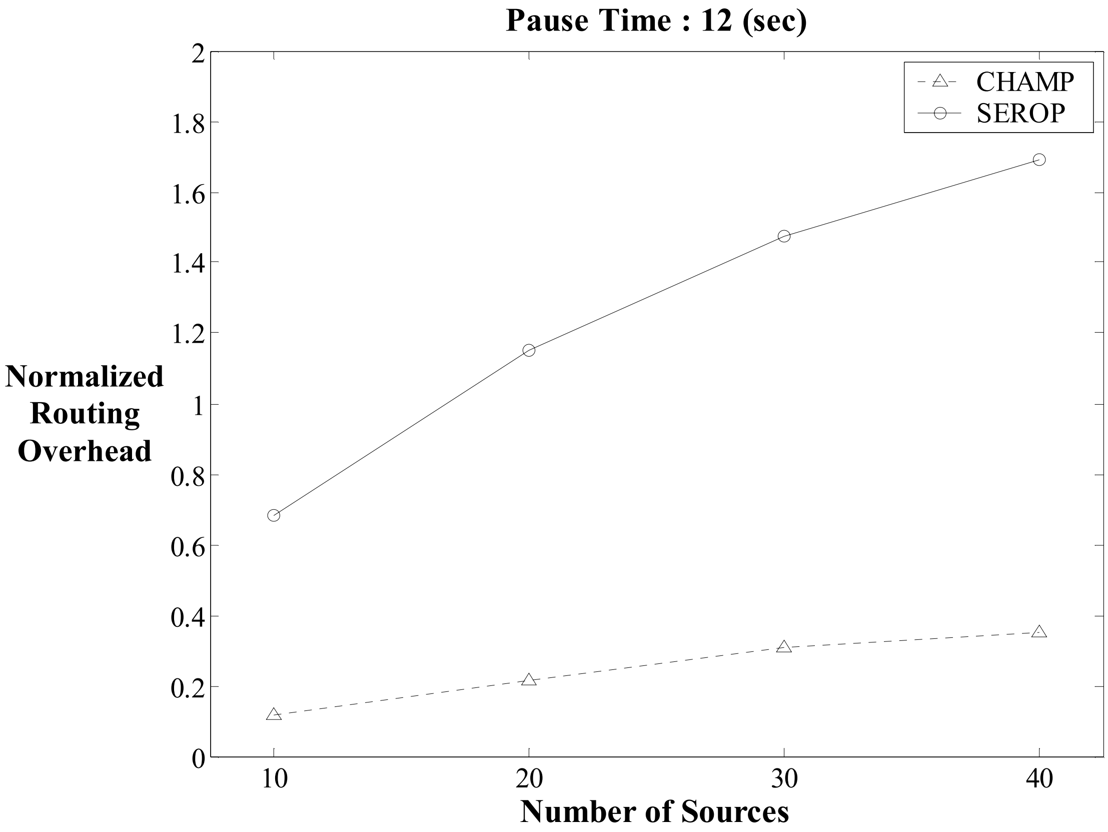
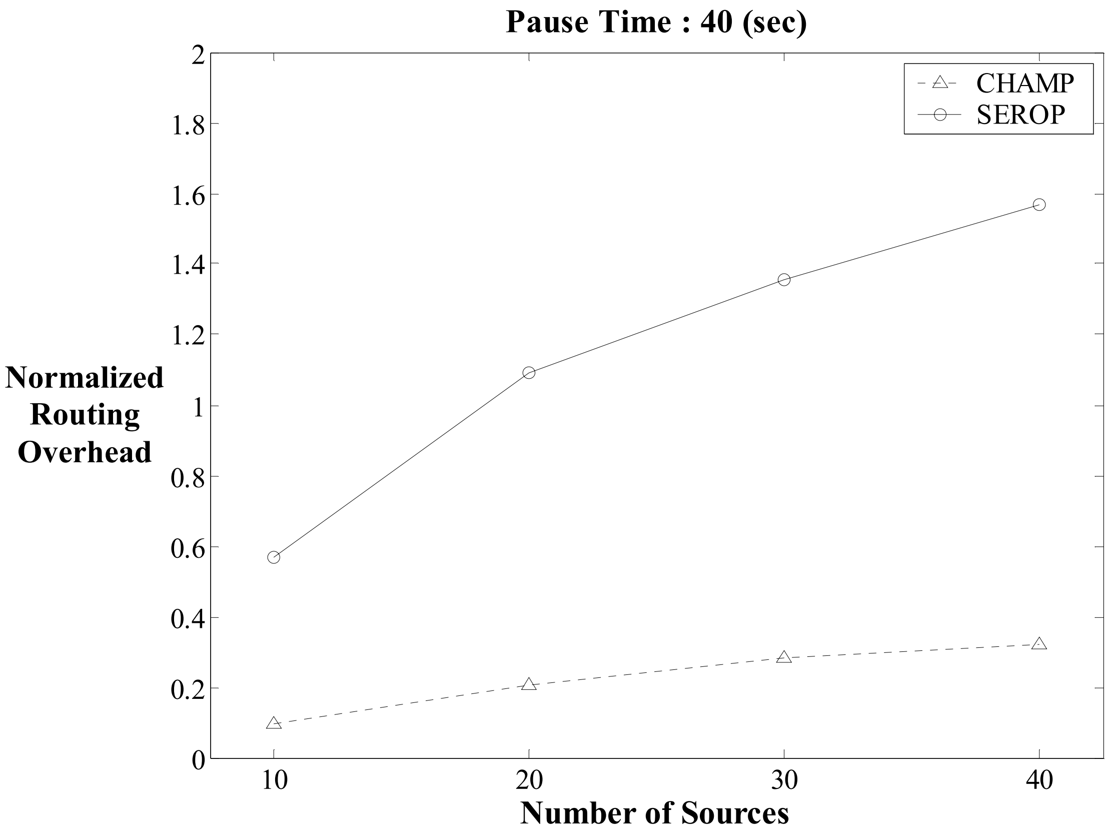


Figure 5.6a Normalized Routing Overhead vs Number of Sources



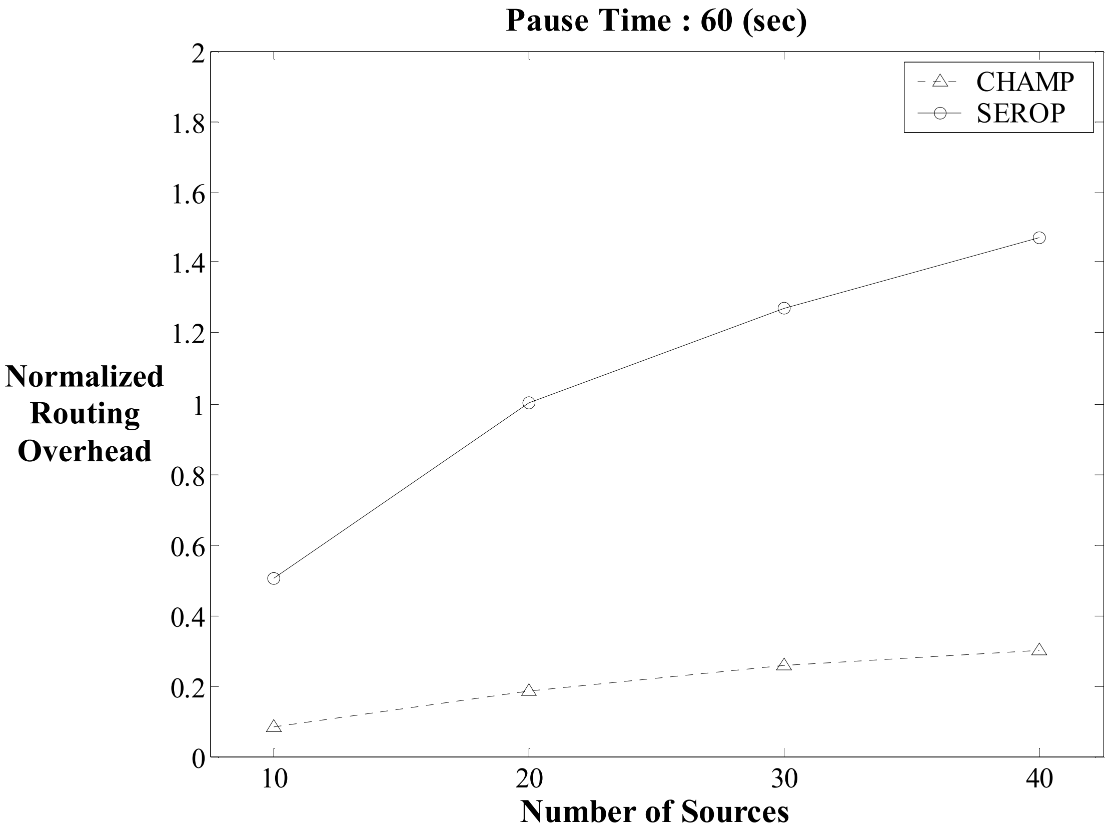
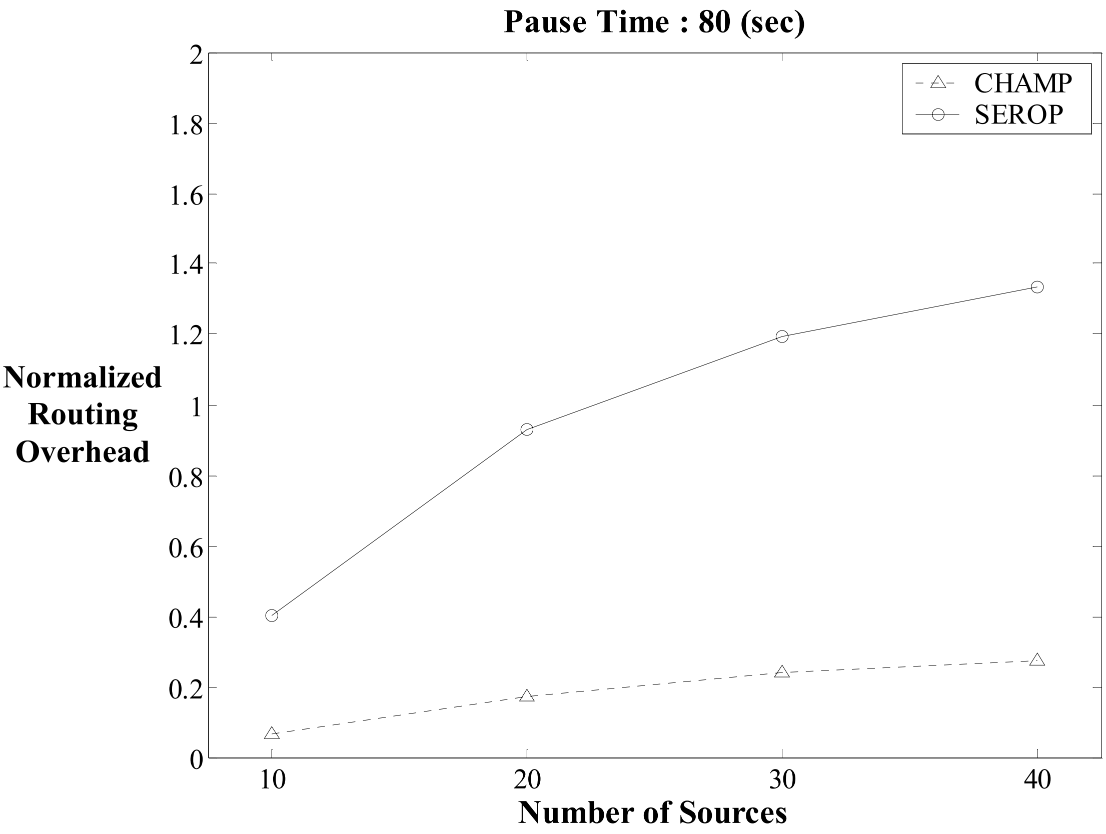


Figure 5.6b Normalized Routing Overhead vs Number of Sources



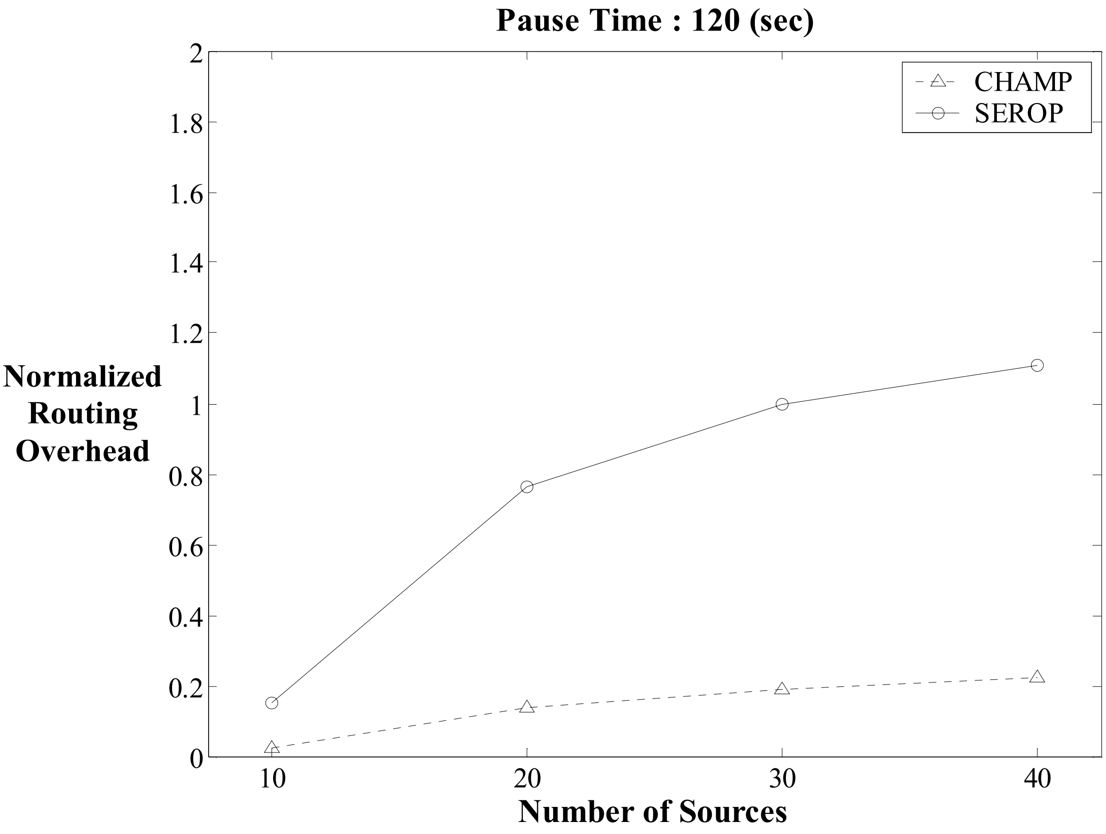


Figure 5.6c Normalized Routing Overhead vs Number of Sources

5.5 Simulation Results for Malicious Network

The simulation results shown in the previous section compare the performance of SEROP with CHAMP in a benign network. In this section, we perform additional runs of simulation to determine the performance of SEROP in the present of malicious nodes. Figure 5.7, 5.9 and 5.11 represent the results of first attack scenarios while figure 5.8, 5.10 and 5.12 illustrate the results of second attack scenarios, as described in the previous section.

In both scenarios, the percentage of packets delivered decreases as the number of malicious nodes increases as depicted in figure 5.7 and figure 5.8. The decreases are due to isolation of malicious nodes that were participating in the routing. However, the results are encouraging when dealing with malicious nodes, especially in second attack scenario. As seen in figure 5.7, SEROP is able to deliver 43.5% of packets successfully at pause time of 120 seconds although 50% of the nodes in the network maliciously tamper the contents of RREQ. Besides that, the PDR at all pause time drops slowly as the percentage of malicious nodes in the network increases from 30% to 50%.

For the second attack scenario, SEROP delivers more than 80% of the data packets at all pause time when 30% of the network nodes are malicious as shown in figure 5.8. Furthermore, SEROP achieves more than 65% of PDR at all pause time even as half of all the nodes in the network rebroadcast the modified RREP or INREP.

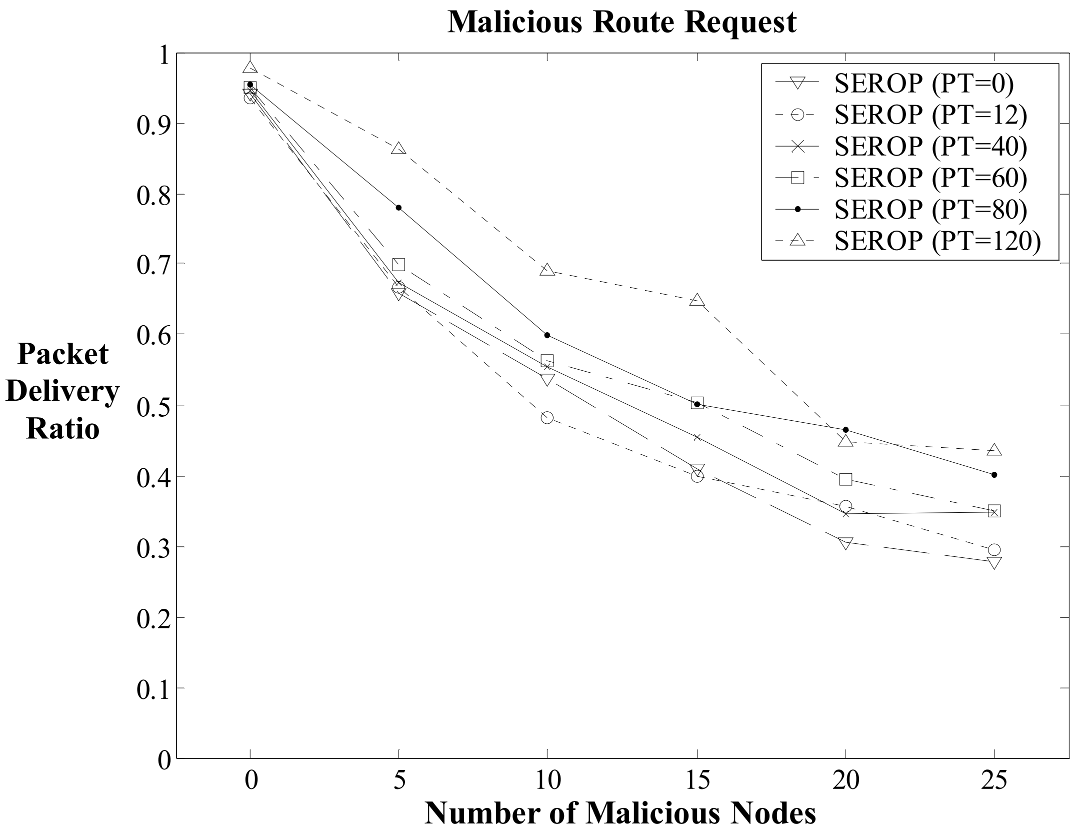


Figure 5.7 Packet Delivery Ratio vs Number of Malicious Nodes (Malicious Route Request)

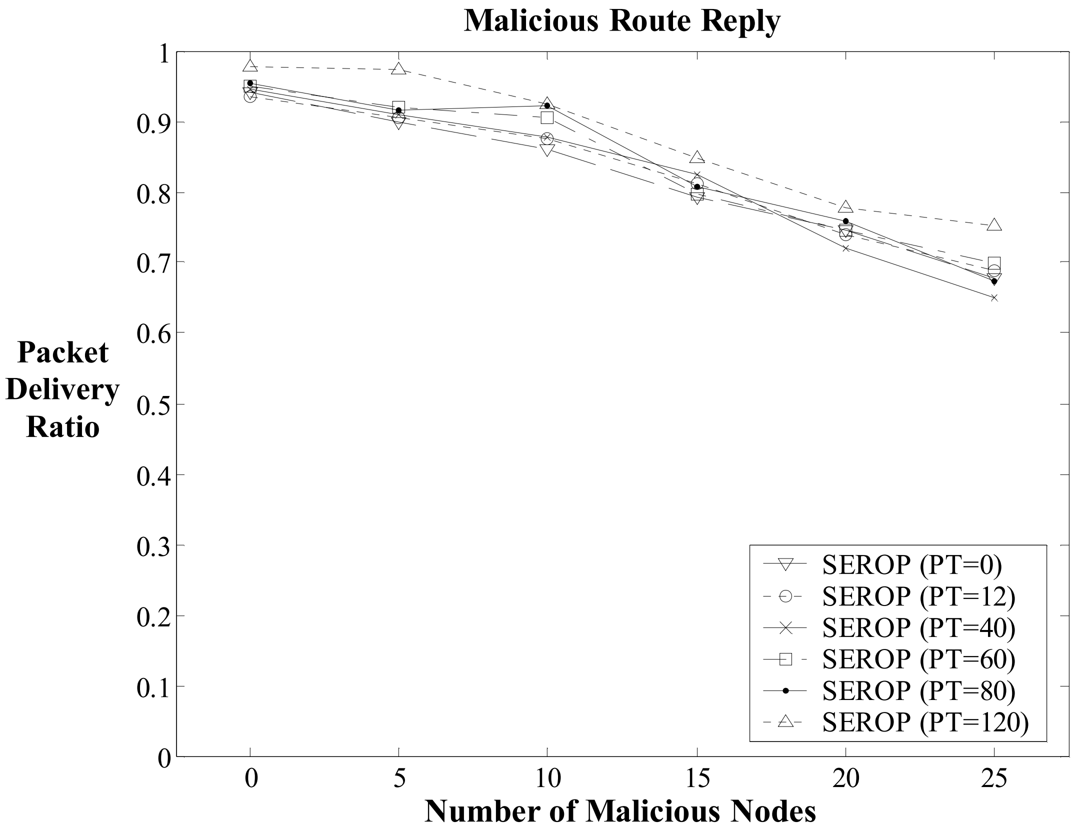


Figure 5.8 Packet Delivery Ratio vs Number of Malicious Nodes (Malicious Route Reply)

End-to-end delay of SEROP increases significantly as the number of malicious nodes increases as depicted in figure 5.9. On the other hand, figure 5.10 illustrates a better delay performance in the second attack scenario as compared to the first attack scenario.

Byte overhead in figure 5.11 is worse than in figure 5.12, due to the frequent regeneration of the RREQ, which has been discarded whenever the authentication fails, and the contribution from the security overhead. However, the byte overhead for the second attack scenario is very low and remains constant even though 50% of the malicious nodes are spread over the entire network, as shown in figure 5.12.

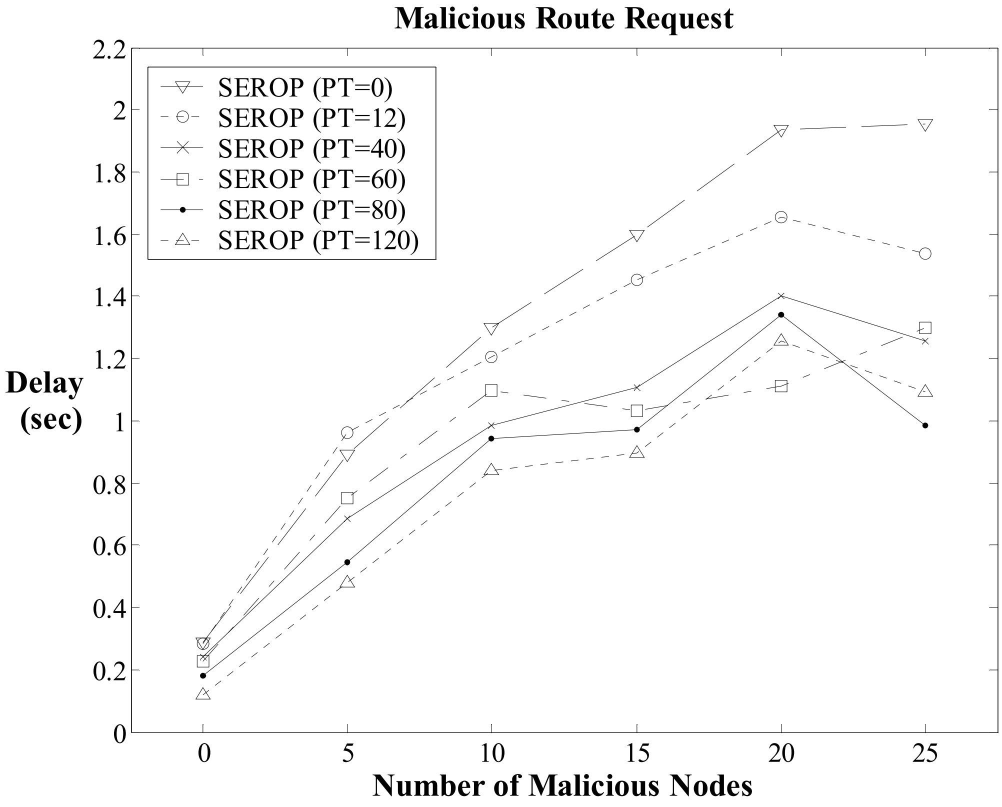


Figure 5.9 End-to-end Delay vs Number of Malicious Nodes (Malicious Route Request)

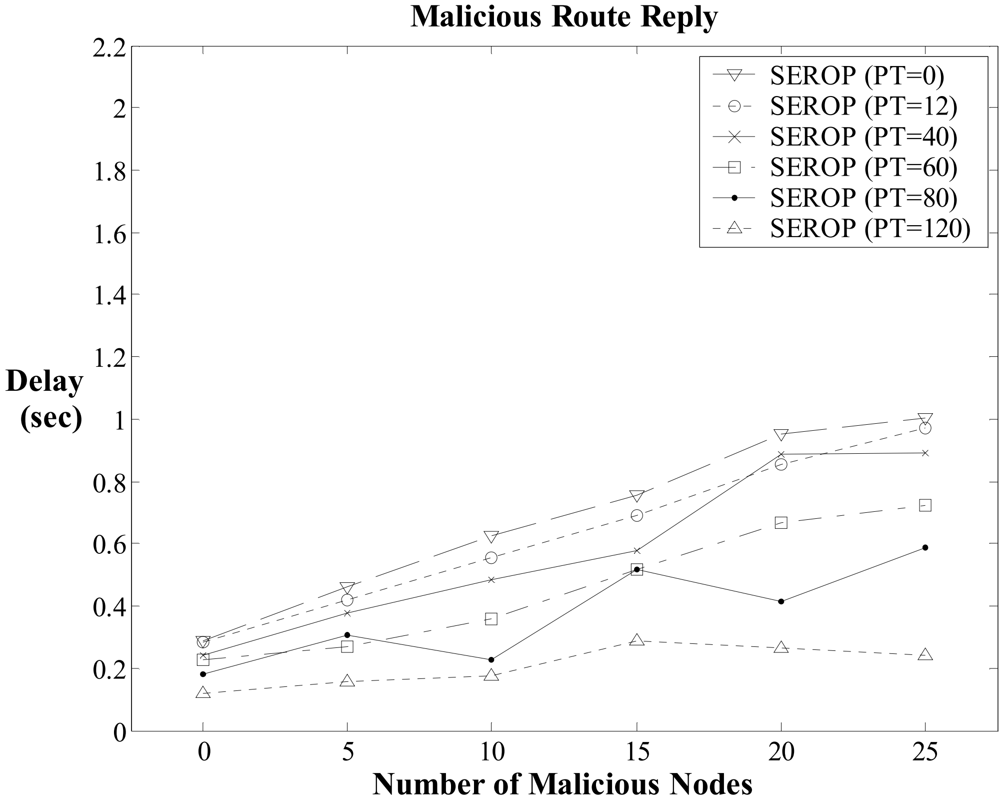


Figure 5.10 End-to-end Delay vs Number of Malicious Nodes (Malicious Route Reply)

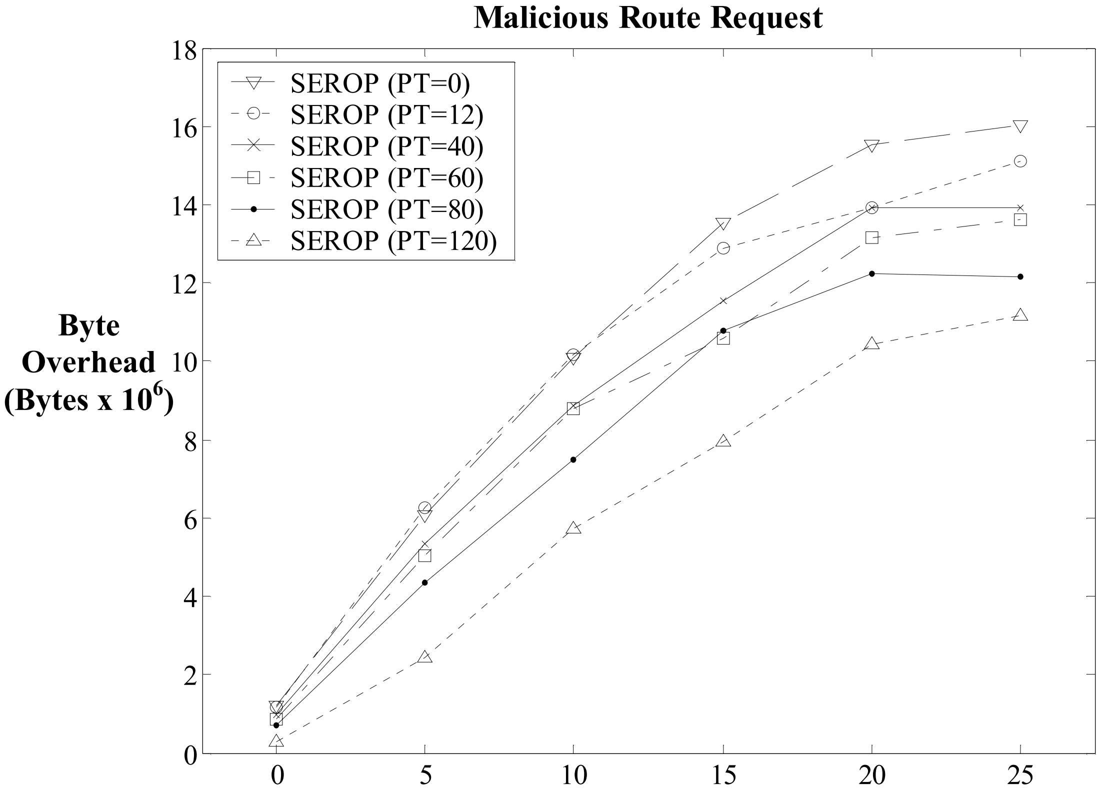


Figure 5.11 Byte Overhead vs Number of Malicious Nodes (Malicious Route Request)

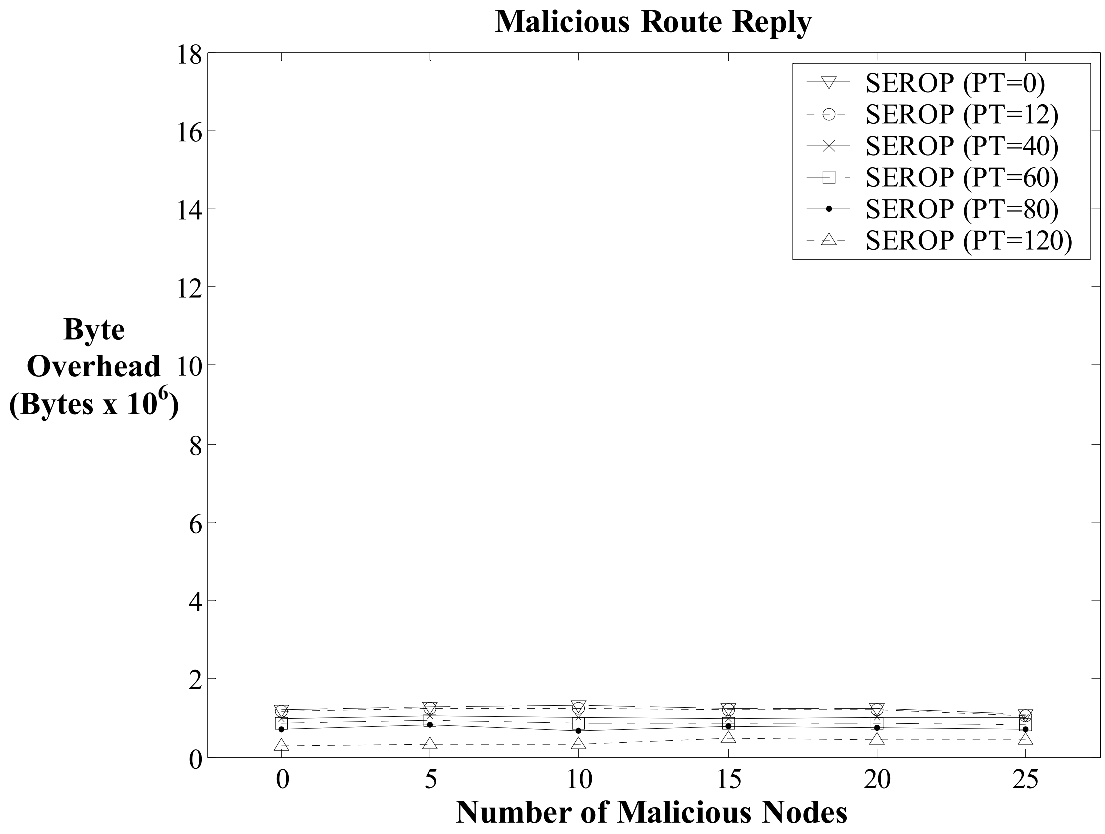


Figure 5.12 Byte Overhead vs Number of Malicious Nodes (Malicious Route Reply)

5.6 Summary

In this section, the performance evaluation of SEROP was presented. The performance of SEROP in benign networks is encouraging. It produces high packet delivery rate, low end-to-end delay and reasonable routing overhead. Besides that, the SEROP protocol is able to function and perform well in the presence of malicious nodes up to as high as 50%. Hence, SEROP can be considered as an efficient and practical security extension that does not significantly degrade the overall performance of the base routing protocol.

Chapter 6

Conclusions and Future Work

After a brief conclusion, we discuss some limitations of our work. We also suggest certain possible extensions to our work. They are some aspects of security that we have not addressed and could serve as good topics for future research.

6.1 Conclusion

The routing protocols proposed for Mobile Ad hoc networks are able to meet the basic requirements like dynamically changing network topologies rather well. However, security issues have been primarily ignored. MANET routing protocols must be secured from the viewpoint of authentication, integrity and privacy. These requirements can be partially met by strong authentication and encryption mechanisms, digital signatures, etc. Moreover, the protection mechanisms can be optimized for every protocol based on the approach taken to routing.

This thesis has presented the design of SEROP, an efficient secure routing protocol (extension) for mobile ad hoc networks that achieves secrecy of data messages and secures the routing operation. The security scheme aims at preventing attacks by malicious nodes that intentionally disrupt the route discovery process. The protocol also assures a source node that generates a route discovery request is able to identify and authenticate the route reply from the destination node. Besides that, the scheme also provides optional secrecy protection for data packet.

For simulation implementation in both benign and malicious networks, we base our protocol, SEROP, on the basic operation of the Caching and Multi-Path (CHAMP) [1] routing protocol. SEROP performs very well in benign networks. It does not reduce the packet delivery ratio of the based protocol by more than 2.7% for 10 data sources throughout the entire simulation. Furthermore, it also exhibits low end-to-end delays and acceptable routing overhead. Besides that, the SEROP protocol is able to function and perform well although 50% of the network nodes are malicious.

Ultimately, SEROP can be regarded as an effective and practical security extension that does not drastically degrade the overall performance of the based routing protocol.

6.2 Limitation

The protocol as compared to symmetric key based schemes, incurs a higher overhead in the network since it uses asymmetric keys to ensure the robustness of the routing protocol. This is usually the trade off between the preferred security levels and the affordable routing overhead. However, this routing overhead can be reduced by using short asymmetric keys if the target scenario is only valid for few hours, e.g. a meeting.

6.3 Future Work

Intrusion detection schemes that analyze traffic profiles to detect intruders would be another challenging area to explore. We have considered isolated cases of node compromise. It would be extremely challenging to study the case where nodes collaborate to bring down the system. Detection of compromised nodes is a very tough problem especially in a dynamically changing network.

Bibliography

[1] Alvin Valera, Winston K.G. Seah, S.V. Rao, "CHAMP: A Highly-Resilient and Energy-Efficient Routing Protocol for Mobile Ad hoc Networks," *Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN 2002)*, Sep 9-11, Stockholm, Sweden.

[2] K. Sanzgiri, B. Dahill, B. N. Levine, C. Shields, and E. M. Belding-Royer, "A Secure Routing Protocol for Ad hoc Networks," In *Proceedings of the* 10*th IEEE International Conference on Network Protocols (ICNP)*, November 2002.

[3] Charles E. Perkins, Elizabeth M. Royer, Samir R. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," *IEEE Internet Draft*, Jan. 2002. draft-ietf-manet-aodv-10.txt (work in progress).

[4] C. R. Davis, *IPSec: Securing VPNs*, McGraw-Hill, New York, NY, USA, 2000.

[5] Corson, S. and Macker, J., "Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations," *RFC2501*, Internet Society, Jan. 1999.

[6] A. Perrig, R. Canetti, D. Tygar, and D. Song, "Efficient Authentication and Signature of Multicast Streams over Lossy Channels," In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 56–73, May 2000.

[7] C. Perkins and P. Bhagwat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," In *Proceedings of the ACM SIGCOMM Conference on Communication Architectures, Protocols, and Applications*, pages 234–244, August 1994.

[8] Giuseppe Ateniese, Michael Steiner, and Gene Tsudik, "New Multiparty Authentication Services and Key Agreement Protocols," *IEEE Journal on Selected Areas in Communication*, 18(4):628-640, April 2000.

[9] Karpijoki, V., "Signalling and Routing Security in Mobile Ad hoc Networks," *Proceedings of the Helsinki University of Technology, Seminar on Internetworking – Ah hoc Networks*, Spring 2000.

[10] Keng Seng NG, and Winston K. G. SEAH, "Routing Security and Data Confidentiality for Mobile Ad hoc Networks," *Proceedings of the 57th IEEE Semiannual Vehicular Technology Conference (VTC2003-Spring)*, IEEE, Jeju, Korea, April 2003.

[11] Lakshmi Venkatraman, "Securing Routing Protocol for Ad hoc Networks" *Technical report, University of Cincinnati*, Nov. 2000.

[12] Lidong Zhou and Zygmunt J. Haas, "Securing Ad hoc Networks," *IEEE Network Magazine*, 13(6), November-December 1999.

[13] F. Stanjo and R. Anderson, "The Resurrecting Duckling: Security Issues in Ad hoc Wireless Networks," Security Protocols, *7th International Workshop Proceedings*, Springer-Verlag 1999.

[14] Manel Guerrero Zapata, "Secure Ad hoc On-Demand Distance Vector (SAODV) Routing," *http://ant.eupvg.upc.es/~tarom/draft-tarom-manet-saodv-00.txt*, Aug. 2001.

[15] N. Asokan and Philip Ginzboorg, "Key-Agreement in Ad hoc Networks," *Elsevier Preprint*, 2000.

[16] R. Rivest, A. Shamir, and L. Adleman: "A Method for Obtaining Digital Signatures and Public-Key Cryptosystems", *Communications of the ACM*, v. 21, n. 2, February 1978.

[17] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker, "Mitigating Routing Misbehavior in Mobile Ad hoc Networking," In *6th International Conference on Mobile Computing and Networking (MOBICOM'00)*, pages 255-265, Aug. 2000.

[18] Yongguang Zhang and Wenke Lee, "Intrusion Detection in Wireless Ad hoc Networks," In *6th International Conference on Mobile Computing and Networking (MOBICOM'00)*, pages 275-283, June 2000.

[19] William Stallings, "Cryptography and Network Security: Principles and Practice, Second Edition," *Prentice Hall International, Inc.*, 1999.

[20] Yih-Chun Hu, D. B. Johnson, and A. Perrig, "Secure Efficient Distance Vector Routing in Mobile Wireless Ad hoc Networks," In *Proceedings of the* 4*th IEEE Workshop on Mobile Computing Systems and Applications* (WMCSA), June 2002.

[21] T. Narten, E. Nordmark, and W. A. Simpson, "Neighbor discovery for IP version 6 (IPv6)", *Internet RFC 2461*, December 1998.

[22] P. Papadimitratos and Z.J. Haas, "Secure Routing for Mobile Ad Hoc Networks," *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, San Antonio, TX, January 27-31, 2002.

[23] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," *Proceedings of the Eighth ACM International Conference on Mobile Computing and Networking (MobiCom 2002)*, ACM, Atlanta, GA, September 2002.

[24] C. Perkins, "Ad Hoc Networking," *Addison-Wesley* 2001

[25] Saab NetDefence, http://www.saab.se/future/node2567.asp.

[26] J. M. Kahn, R. H. Katz and K. S. J. Pister, “Next Century Challenges: Mobile Networking for Smart Dust,” *ACM Press* 1999.

[27] A. Shamir, “How to Share a Secret,” *Communications of ACM*, 1979.

[28] L. Gong, R. Needham, and R.Yahalom, “Reasoning about Belief in Cryptographic Protocol,” *Proceedings of the 1990 IEEE Symposium on Research in Security and Privacy*,” p. 234-248, IEEE Computer Society Press, Los Alamitos, CA, 1990.

[29] M.Burrows, M. Abadi, and R.Needham, “A Logic of Authentication,” in *Proceedings of 12th ACM Symposium on Operating Systems Principles*, Arizona, Dec. 1989.

[30] Seung Yi, Prasad Naldurg, Robin Kravets, “Integrating Quality of Protection into Ad Hoc Routing Protocols,” *The 6th World Multi-Conference on Systemics, Cybernetics and Informatics (SCI 2002),* 2002.

[31] Z. Yu, T. Jiang, X. Wu and W. A. Arbaugh, “Risk Based Probabilistic Routing for Ad-Hoc Networks,” poster in *ACM Workshop on Wireless Security (WiSe)* 2002, Atlanta, U.S.A.

[32] Weichao Wang, Yi Lu, Bharat Bhargava, “On Vulnerability and Protection of Ad Hoc On-demand Distance Vector Protocol,” *Proceedings of International Conference on Telecommunication (ICT’2003)*, France, February 2003.

[33] Weichao Wang, Yi Lu, Bharat Bhargava, “On Security Study of Two Distance Vector Routing Protocols for Mobile Ad Hoc Networks,” *the first IEEE Annual Conference on Pervasive Computing and Communications (PerCom’2003),* Dallas-Fort Worth, Texas, March 2003.

[34] Joo-Han Song, Yoji Kawamoto, Vincent Wong, Victor Leung, “Secure Routing with Tamper Resistant Module for Mobile Ad Hoc Networks,” (poster)

MobiHoc 2003 The Fourth ACM International Symposium on Mobile Ad Hoc Networking and Computing Annapolis, Maryland, USA June 1-3, 2003.

[35] Seungjoon Lee, Bohyung Han, Minho Shin, "Robust Routing in Wireless Ad Hoc Networks," *2002 International Conference on Parallel Processing Workshops (ICPPW'02)* August 18 - 21, 2002 Vancouver, B.C., Canada.

[36] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, "Efficient Security Mechanisms for Routing Protocols," *Proceedings of the Tenth Annual Network and Distributed System Security Symposium (NDSS 2003)*, ISOC, San Diego, CA, February 2003.

[37] Stephen Carter and Alec Yasinsac, "Secure Position Aided Ad hoc Routing Protocol," *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02),* Nov 3-4, 2002.

[38] S. Bhargava and D. P. Agrawal, "Security enhancements in aodv protocol for wireless ad hoc networks," *Vehicular Technology Conference*, 2001.

[39] Alec Yasinsac Vikram Thakur Stephen Carter Ilkay Cubukcu, "A Family of Protocols for Group Key Generation in Ad Hoc Networks," *Proceedings of the IASTED International Conference on Communications and Computer Networks (CCN02)*, Nov 3-4, 2002.

[40] Seung Yi, Robin Kravets, "MOCA: Mobile Certificate Authority for Wireless Ad Hoc Networks," *2nd Annual PKI Research Workshop Program (PKI 03)*, Gaithersburg, Maryland, April, 2003.

[41] M. C. Morogan and S. Muftic, "Certificate Management in Ad hoc Networks," *IEEE Workshop on Security and Assurance in Ad hoc Networks*, in conjunction with the 2003 International Symposium on Applications and the Internet, Orlando, FL, January 28, 2003.

[42] D.W. Carman, B.J. Matt and G.H. Cirincione, "Energy-efficient and Low-latency Key Management for Sensor Networks," *In Proceedings of 23rd Army Science Conference*, Dec 2-5 2002 Orlando Florida.

[43] Srdjan Capkuny, Levente Buttyan and Jean-Pierre Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks," *Swiss Federal Institute of Technology Lausanne (EPFL) Tech. Report* (Jun 2002).

[44] A. Khalili, J. Katz and W. Arbaugh, "Toward Secure Key Distribution in Truly Ad-Hoc Networks," *IEEE Workshop on Security and Assurance in Ad hoc Networks*, in conjunction with the 2003 International Symposium on Applications and the Internet, Orlando, FL, January 28, 2003.

[45] Silja Mäki, Tuomas Aura, Maarit Hietalahti, "Robust Membership Management for Ad-hoc Groups," *In Proceedings of the 5th Nordic Workshop on Secure IT Systems (NORDSEC 2000).*

[46] Alec Yasinsac Jim Davis, "Modeling Protocols for Secure Group Communication in Ad Hoc Networks," (Extended Abstract) *Tenth International Workshop on Security Protocols*, Cambridge, UK, Apr 17-19, 2002, LNCS.

[47] Adrian Perrig, Robert Szewczyk, Victor Wen, David Culler, J. D. Tygar, "SPINS: Security Protocols for Sensor Networks," in *Wireless Networks Journal (WINE)*, September 2002.

[48] Lin Yuan, Gang Qu, "Design Space Exploration for Energy-Efficient Secure Sensor Network," *The IEEE International Conference on Application-Specific Systems, Architectures, and Processors (ASAP'02)* July 17 - 19, 2002 San Jose, California.

[49] Donggang Liu and Peng Ning, "Efficient Distribution of Key Chain Commitments for Broadcast Authentication in Distributed Sensor Networks,"

The 10th Annual Network and Distributed System Security Symposium, San Diego, California. February 2003.

[50] Jing Deng, Richard Han, and Shivakant Mishra, “A Performance Evaluation of Intrusion-Tolerant Routing in Wireless Sensor Networks,” *2nd International Workshop on Information Processing in Sensor Networks (IPSN '03),* Palo Alto, CA, USA, April, 2003.

[51] Loukas Lazos and Radha Poovendran, “Secure Broadcast in Energy-Aware Wireless Sensor Networks,” (Invited Paper), *IEEE International Symposium on Advances in Wireless Communications (ISWC'02)*, September 23-24, 2002, Victoria, BC, Canada.

[52] Candolin, H. Kari, “A Security Architecture for Wireless Ad hoc Networks,” *IEEE Milcom 2002*, Anaheim, California, October 7-10, 2002.

[53] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, “Packet Leashes: A Defense against Wormhole Attacks in Wireless Ad Hoc Networks,” *Proceedings of the Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2003)*, IEEE, San Francisco, CA, April 2003.

[54] Haiyun Luo, Jiejun Kong, Petros Zerfos, Songwu Lu and Lixia Zhang, “Self-securing Ad Hoc Wireless Networks” accepted by the *Seventh IEEE Symposium on Computers and Communications (ISCC'02).*

[55] Jiejun Kong, Haiyun Luo, Kaixin Xu, Daniel Lihui Gu, Mario Gerla, Songwu Lu, “Adaptive Security for Multi-layer Ad-hoc Networks” issue in *John Wiley InterScience Press journal* "Special Issue of Wireless Communications and Mobile Computing". August, 2002.

[56] M. Mathis, J. Mahdavi, S. Floyd, and A. Romanow, “TCP selective acknowledgment options”, *Internet RFC 2018*, October 1996.