# Load-Balancing and Secure Routing for Wireless Mobile Ad Hoc Networks

by

JOO-HAN SONG

M. Sc., Hongik University, 2001

B. Sc., Hongik University, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF
DOCTOR OF PHILOSOPHY

in

THE FACULTY OF GRADUATE STUDIES
ELECTRICAL AND COMPUTER ENGINEERING

THE UNIVERSITY OF BRITISH COLUMBIA

March 2005

# Abstract

A mobile ad hoc network is a collection of mobile nodes dynamically forming a multi-hop wireless network. Due to the limited transmission range of wireless interface, a routing protocol is needed to enable communication between mobile nodes. The first part of our work focuses on the load-balancing routing protocol in *mobile ad hoc wireless access network* in which mobile nodes can access the Internet via one or more stationary gateway nodes. Although on-demand routing schemes are appealing with low routing overhead in bandwidth restricted networks, their routing control overhead increases exponentially with node density in a given geographic area. To control the overhead of on-demand routing without sacrificing performance, we present a novel extension of the Ad hoc On-demand Distance Vector (AODV) routing protocol, called LB-AODV, which incorporates the concept of load-balancing (LB). Results show that LB-AODV outperforms other routing schemes as traffic increases. We compare the performance of our proposed LB-AODV protocol with both the original AODV and gossip-based routing protocols in different mobility and traffic scenarios. Simulation results show that LB-AODV delivers more data packets to the gateway and decreases the end-to-end delay of packets delivered by reducing the transmissions of routing control messages by 50 % or more. In scenarios with traffic congestion, LB-AODV outperforms AODV and GOSSIP1 routing protocols.

Another important issue in ad hoc routing protocol is security. In hostile environments, it is crucial to provide a secure communication infrastructure between mobile nodes. We investigate the severe threats of *routing table tampering attack*, where attackers can modify the information stored in a mobile host's routing table. To guard against this attack, we propose the use of either the Tamper Resistant Module (TRM) or the Secure Table Entry Protection (STEP)

mechanisms. STEP can provide the authentication for both the destination sequence number and hop-count fields in the routing table entry. We analyze security threats against AODV routing protocol and propose Secure AODV (SeAODV), a secure routing extension to the original AODV. We also propose a Secure Data Forwarding (SDF) scheme based on SeAODV for secure transmissions of data packets over the wireless links. Simulation results showed that STEP continues to maintain a high packet delivery fraction and a small end-to-end delay at the expense of slightly higher route acquisition latency and control overhead in route discovery. In the presence of either data packet dropping or routing table tampering attacks, SeAODV continues to maintain a high packet delivery fraction and a small end-to-end delay.

We further investigate the security problems of position-based routing protocol in mobile ad hoc networks. Although position-based routing protocols can offer significant performance improvement over topology-based routing protocols by using location information in large and dense mobile ad hoc networks, attackers can disrupt the location service by violating protocol specifications. We propose the Secure Grid Location Service (SGLS), which is an enhancement to the original Grid Location Service (GLS) protocol by using a broadcast authentication protocol and a Local Reputation System (LRS) for monitoring. Simulation results showed that in the presence of message dropping attacks, the proposed LRS mechanism maintains a high message delivery ratio at the expense of a higher average end-to-end delay and routing overhead.

# Table of Contents

# List of Tables

# List of Figures

# List of Acronyms

| | |
|---|---|
| ACK | Acknowledgement |
| AODV | Ad hoc On-demand Distance Vector |
| ARAN | Authenticated Routing for Ad hoc Networks |
| CA | Certification Authority |
| CBR | Constant Bit Rate |
| CONFIDANT | Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks |
| CORE | COllaborative REputation mechanism |
| CRL | Certificate Revocation List |
| CTS | Clear To Send |
| DCF | Distributed Coordinated Function |
| DH | Diffie-Hellman |
| DIFS | DCF Inter Frame Space |
| DoS | Denial of Service |
| DREAM | Distance Routing Effect Algorithm for Mobility |
| DSR | Dynamic Source Routing |
| ECC | Elliptic Curve Cryptography |
| EEPROM | Electrically Erasable Programmable Read-Only Memory |
| ESTEP | Efficient Secure Table Entry Protection |
| GLS | Grid Location Service |
| GPS | Global Positioning System |
| HMAC | Hashed Message Authentication Code |
| ID | Identifier |
| IETF | Internet Engineering Task Force |

| | |
|---|---|
| IP | Internet Protocol |
| LB | Load-Balancing |
| LE | Location Error |
| LI | Location Information |
| LQ | Location Query |
| LR | Location Reply |
| LRS | Local Reputation System |
| LU | Location Update |
| MANET | Mobile Ad hoc Network |
| MD | Message Digest |
| NIC | Network Interface Card |
| PLCP | Physical Layer Convergence Protocol |
| RERR | Route Error |
| RFC | Request For Comments |
| RREP | Route Reply |
| RREQ | Route Request |
| RSA | Rivest Shamir Adleman |
| RTS | Request To Send |
| SAODV | Secure AODV |
| SDF | Secure Data Forwarding |
| SEAD | Secure Efficient Ad hoc Distance vector |
| SeAODV | Secure AODV |
| SGF | Secure Geographic Forwarding |
| SGLS | Secure Grid Location Service |
| SIFS | Short Inter Frame Space |

| | |
|---|---|
| SLSP | Secure Link-State Protocol |
| SRP | Secure Routing Protocol |
| STEP | Secure Table Entry Protection |
| TESLA | Timed Efficient Stream Loss-tolerant Authentication |
| TIK | TESLA with Instant Key disclosure |
| TRH | Tamper Resistant Hardware |
| TRM | Tamper Resistant Module |
| TRS | Tamper Resistant Software |
| UDP | User Datagram Protocol |
| VHR | Virtual Home Region |
| ZRP | Zone Routing Protocol |

# Acknowledgements

I am indebted to many individuals for their care and support given to me during my graduate studies. First, I would like to express my deep gratitude to Professor Victor Leung and Professor Vincent Wong, for their wisdom and guidance during my years as a graduate student. As my supervisors, they have provided me constant encouragement, insightful comments, and invaluable suggestions. I greatly appreciate all the freedom they gave me to pursue whatever research problems interested me. I am also grateful to them for allowing me to attend and present our work at various conferences.

I would like to thank Professors Cyril Leung and Hussein Alnuweiri for serving on my dissertation committee. I would like to thank Professors Son Vuong and Mabo Ito for serving as university examiners. I would also like to thank Professor Sherman Shen from the University of Waterloo for serving as external examiner. I am thankful for the opportunity to work with Mr. Yoji Kawamoto from Sony Corporation in Japan during his visit at UBC. Yoji provided motivations and insights on our work of secure routing protocols.

I am also grateful to the former and present members of the Communications group for their friendship, help, and support, especially Farshid Agharebparast. I would also like to thank my friends, who helped me in many ways through good times and bad.

I wish to give special thanks to my parents who always have loved me, believed in me, and encouraged me in my study. Finally, many thanks should go to my wife, Weejung, and my two sons, Woohyuk and Woosung. I could not complete my graduate studies without their dedicated sacrifice, understanding, and encouragement.

This work was supported in part by the (1) Natural Sciences and Engineering Research Council of Canada (NSERC) under grants PGPIN 261604-03 and 44286-00; and (2) the University of British Columbia under the University Graduate Fellowship.

# Co-Authorship Statement

I am the first author and principle contributor of all manuscript chapters. All manuscript chapters are co-authored with V.W.S. Wong and V.C.M. Leung, who co-supervised the thesis research. Chapter 3 is also co-authored with Y. Kawamoto who contributed the idea of using temper resistant modules.

# Chapter 1 Introduction

The advancement in portable computing technologies and the emergence of mobile/nomadic applications have generated a lot of interest in wireless network infrastructures. A Mobile Ad hoc NETwork (MANET) [1] consists of a set of wireless mobile nodes communicating with each other without any centralized control or fixed network infrastructure. Each mobile host serves as a router having the capability to perform packet forwarding for other mobile nodes that may not be within direct wireless transmission range of each other. MANETs have been evolving to serve a growing number of applications that rely on multi-hop wireless infrastructures that can be deployed quickly. The potential applications include emergency disaster relief, battlefield command and control, mine site operations, and wireless classrooms or meeting rooms in which participants wish to share information or to acquire data.

Today, advances in wireless technologies such as IEEE 802.11 [2] wireless local area networks (WLANs), Bluetooth [3] and third generation cellular networks have led to a proliferation of mobile devices. The number of mobile devices is expected to reach a billion in the near future [4] and exceed the number of stationary nodes. We expect that in future MANETs will be interconnected to the Internet in some applications. We consider a mobile networking environment in which mobile hosts can access the Internet directly via one or more gateways or access points, or indirectly via other mobile hosts. This is referred as a *mobile ad hoc wireless access network* or *wireless mesh network* [5]. Mobile hosts that are near the gateway can communicate directly with the gateway via single hop connections. Mobile hosts that are outside the transmission range of the gateway have to use multi-hop connections that rely on the neighboring mobile nodes to relay their packets (see Figure 1.1). Commercial applications include accessing the Internet through multi-hop wireless links. Practical examples of these networks include wireless mesh networks [5], sensor networks [6], and rooftop networks [7].

Moreover, recently the IEEE has approved the mesh project, IEEE P802.11s [8], which extends the coverage area of WLAN by allowing data to pass through wireless nodes. In addition to providing Internet access, this network configuration may also serve other practical scenarios; e.g., the gateways may represent nodes that host special services such as domain name service accessed by other nodes in the MANET.

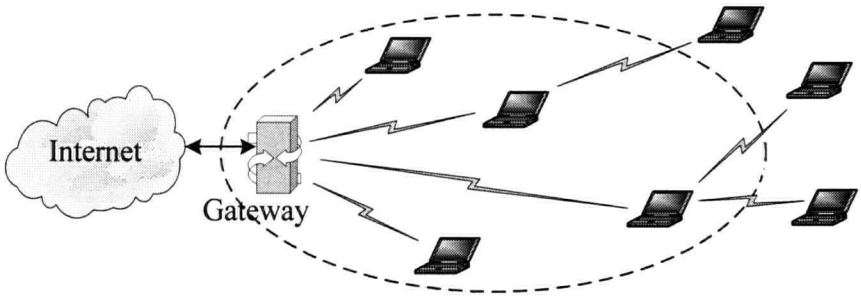



Figure 1.1 Mobile ad hoc wireless access network.

One of the issues in ad hoc routing protocols is to *reduce the routing overhead without degrading the network connectivity*. There is a trade-off between maintaining the full network connectivity and minimizing bandwidth contention in a MANET. Although increasing the total number of mobile nodes tends to reduce the effective bandwidth available at individual nodes due to increased competition for bandwidth in a given coverage area, it also increases the connectivity of the network, which may be important as node mobility increases. Results in [9]–[11] show that when the number of nodes is small, the network may not be fully connected (i.e., some nodes may not be able to send packets to certain destinations). Network connectivity can be increased by simply increasing the total number of mobile nodes in the network. However when the number of nodes and the traffic load increase, contention and packet collisions between neighboring nodes also increase exponentially [12].

Another issue in ad hoc routing protocol is *to secure the communication between*

*mobile nodes in a hostile environment.* Most of the current ad hoc routing protocols [13][14] proposed for MANETs assume that there is an implicit *trust-your-neighbor* relationship in which all the neighboring nodes behave properly. However, in practice MANETs may be subject to attacks by rogue users, who try to paralyze the networks by manipulating the messages (e.g., dropping all data or control packets, sending incorrect route advertisement messages). This problem is further complicated by the lack of centralized management control, error-prone multi-hop wireless channels, and the dynamic changes in network topology due to node mobility. Security mechanisms are necessary to prevent a routing table tampering attack and to secure both Ad hoc On-demand Distance Vector's (AODV) [13] control and data messages in MANETs. AODV is currently published [13] as a Request For Comments (RFC) by the Internet Engineering Task Force (IETF).

Current research on MANET has mainly focused on *topology-based* routing protocols, including both proactive and reactive (on-demand) approaches [15]. When either the topology of the network changes frequently or the size of network increases, some of these protocols may incur a significant amount of routing control overhead. Recent research has shown that position-based routing protocols are good alternatives to topology-based routing protocols in large and dense MANETs [16]. In *mobile ad hoc wireless access networks* where the location of the gateway is fixed and known, each source node can send messages to the gateway without introducing any routing control overhead by using position-based routing protocols. Position-based routing protocols avoid the flooding of control traffic by using location information. For an intermediate node to make a packet forwarding decision, it only needs to know its own position and the positions of its neighboring nodes. The message is forwarded to a neighbor geographically closest to the position of the message's destination by using the message forwarding strategies [17]–[19]. To implement a position-based routing protocol, information about the physical location of each destination must be available. Each node can determine its

own position using the Global Positioning System (GPS) [20]. In addition, a *location service* is used by the sender to determine the location of the destination. Each node may have a *location table* to store the position information of other nodes. In a hostile environment, an attacker can disrupt the location service by modifying control messages. Security mechanisms are crucial for both control and data messages in position-based routing protocols.

The rest of this chapter is organized as follows: Section 1.1 discusses the motivations and objectives of our work. Section 1.2 presents an overview of our contributions. Section 1.3 describes the organization of this thesis.

## 1.1 Motivations and Objectives

Since on-demand routing protocols (e.g., AODV [13], Dynamic Source Routing (DSR) [14]) use the flooding method to find a route to the destination, the number of rebroadcasts of an RREQ is proportional to the number of nodes. Therefore, the routing control overhead increases with the total number of nodes. Consider the situation where mobile node $A$ broadcasts an RREQ message to $n$ neighboring nodes. The $n$ neighbors may rebroadcast this RREQ message to their respective neighbors. Packet collisions may occur over the wireless medium, resulting in congestion and possible loss of routing control packets. Furthermore, the source node may attempt to recover from loss of routing control packets by initiating another route discovery process, which further increases the amount of control traffic in the network [21]–[23]. In order to maintain a high packet delivery fraction and a low end-to-end delay for packet transmissions over a MANET, it is important to reduce the amount of routing control traffic [24][25]. Recent research [9] has reported that the best performance can be achieved when the average number of neighbors is around "seven" without degrading the network connectivity.

The above discussion motivates us to design an efficient routing mechanism, which can find a route to the gateway with a controlled amount of routing overhead in *mobile ad hoc wireless access network*. To design an efficient load-balancing routing mechanism, several

factors should be considered including: (1) the number of mobile nodes, (2) the number of source nodes, (3) the size of network, (4) the number of gateways, and (4) the transmission range of wireless interface.

Another part of our work focuses on providing a secure communication between mobile nodes in a hostile environment. Although various secure routing protocols have been proposed to prevent several attacks (e.g., message tampering, message dropping, message replay), the possible threats of a *routing table tampering attack* [26][27] have not been resolved yet. Routing table tampering attacks include the physical deletion, alteration, or falsification of information stored in the routing tables in a node. The objectives of this work are to provide security mechanisms to prevent a routing table tampering attack and to secure both control and data messages in MANETs. We focus on AODV as the basis of the design of our security mechanisms because (1) it is a standardized routing protocol within the IETF and (2) the on-demand distance vector property makes this protocol vulnerable to several malicious attacks [26]–[28]. Our security solution can also be applied to the proposed load-balancing routing mechanism in *mobile ad hoc wireless access networks*.

We further investigate the security problems of position-based routing protocols. Position-based routing protocols are vulnerable to several attacks that are different from those against topology-based routing protocols. For example, in position-based routing protocols, each node may have a *location table* to store the position information of other nodes instead of *routing table*. The sender uses a *location service* to determine the location of the destination. If the position of the destination nodes cannot be determined, then the position-based routing protocols will not function properly. Attackers can also consume the network resources and the node energy by either injecting or dropping messages.

The above discussion motivates us to design security mechanisms for both data and control packets based on the analysis of security threats against position-based routing protocols.

Attackers may continue to launch attacks if there is no penalty or punishment for their misbehaviors. To this end, we also propose a reputation system to detect and isolate attackers.

## 1.2 Main Contributions

The main contributions of this thesis are as follows:

- **Design of an efficient on-demand routing protocol for mobile ad hoc wireless access networks**: We propose a distributed grouping mechanism, which divides the mobile nodes logically into different groups to reduce and distribute routing traffic over the network. We also propose a novel extension of the AODV routing protocol where the route selection is regulated by this grouping mechanism. Our proposed load balancing AODV (LB-AODV) protocol can find a route to the gateway with a controlled amount of routing overhead.
- **Design of secure AODV routing and data forwarding mechanisms:** We analyze security threats and requirements for AODV routing protocol. In order to defend against the threats of routing table tampering attack, we propose the use of a Tamper Resistant Module (TRM) in each mobile node to protect the routing module from attackers. As an alternative method, we propose the Secure Table Entry Protection (STEP) scheme to provide the authentication for both the destination sequence number and hop-count fields in the routing table entry. We also propose Secure AODV (SeAODV), a secure routing extension to the original AODV protocol, to protect routing control messages. We also propose a Secure Data Forwarding (SDF) scheme based on SeAODV for secure transmissions of data packets over the wireless links by maintaining the integrity of data messages.
- **Design of secure position-based routing protocol**: We analyze the security threats against position-based routing protocols in mobile ad hoc networks. In light of these threats, we describe the security requirements for position-based routing protocols. We propose a Secure Geographic Forwarding (SGF) mechanism, which provides *message authentication* by using

both the shared key and the broadcast authentication method with tight time synchronization. In combination with SGF, we propose a Secure Grid Location Service (SGLS) where any receiver can verify the correctness of location messages. To detect and isolate message dropping attackers, we propose the use of a Local Reputation System (LRS) be integrated with the Grid Location Service (GLS) [29].

- **Implementation of proposed schemes in Network Simulator (*ns*2) and performance evaluations by simulations:** We implemented our proposed routing protocols [i.e., LB-AODV, STEP, SeAODV, SDF, SGF, SGLS, and LRS] in either ns-2 version b8a [30] or ns-2 grid package [31], and compare the performance with other routing protocols based on simulations.

## 1.3 Organization of the Thesis

The remainder of this thesis is organized as follows. In Chapter 2, we propose a load-balancing routing scheme for mobile ad hoc wireless access networks. To control the overhead of on-demand routing without sacrificing performance, we present a LB-AODV routing protocol, which incorporates the concept of load-balancing. We present performance comparisons among LB-AODV, AODV, and gossip-based routing protocols. In Chapter 3, we propose the use of both the Tamper Resistant Module (TRM) and the Secure Table Entry Protection (STEP) mechanism to prevent routing table tampering attacks. We also propose a secure routing extension (SeAODV) and secure data forwarding (SDF) mechanism for the AODV routing protocol. We present the performance analysis of proposed scheme with and without attackers. In Chapter 4, we identify the security threats and analyze the security requirements for position-based routing protocols. In consideration of these requirements, we propose a Secure Geographic Forwarding (SGF) mechanism. By combining SGF with the Grid Location Service (GLS), we propose the Secure Grid Location Service (SGLS). We also propose a Local Reputation System (LRS) to detect and isolate misbehaving neighboring nodes. We present the performance analysis of both

SGLS and LRS, and compare them with the original GLS. Chapter 5 concludes the thesis with a summary of our presented work, and describes some directions for future research.

## Bibliography


[1] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," *IETF RFC* 2501, Jan. 1999.

[2] IEEE Computer Society LAN/MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11," Sept. 1999.

[3] Bluetooth Core Specification, version 1.2, Nov. 2003.

[4] C.E. Perkins, J.T. Malinen, R. Wakikawa, A. Nilsson, and A. J. Tuominen, "Internet connectivity for mobile ad hoc networks," *Wiley InterScience Journal of Wireless Communications and Mobile Computing*, vol. 2, issue 5, pp. 465-482, Aug. 2002.

[5] B. Schrick and M. Riezenman, "Wireless broadband in a box," *IEEE Spectrum*, pp. 38-43, June 2002.

[6] I. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "A survey on sensor networks," *IEEE Communications Magazine*, vol. 40, issue 8, pp. 102-114, Aug. 2002.

[7] D. Beyer, M.D. Vestrich, and J.J. Garcia-Luna-Aceves, "The Rooftop Community Network: Free, High-Speed Network Access for Communities," *The First 100 Feet: Options for Internet and Broadband Access.* Editors: D. Hurley and J. H. Keller. MIT Press: Cambridge, 1999.

[8] http://standards.ieee.org/announcements/pr_80211rs.html

[9] E.M. Royer, P.M. Melliar-Smith, and L.E. Moser, "An analysis of the optimum node density for ad hoc mobile networks," in *Proc. of IEEE International Conference on Communications (ICC)*, Helsinki, Finland, pp. 857-861, June 2001.

[10] M. Sanchez, P. Manzoni, and Z.J. Haas, "Determination of critical transmission range in ad hoc networks," in *Proc. of Multiaccess Mobility and Teletraffic for Wireless Communications Workshop (MMT)*, Venice, Italy, pp. 293-304, Oct. 1999.

[11] L. Kleinrock and J. Silvester, "Optimum transmission radii for packet radio networks or why six is a magic number," in *Proc. of IEEE National Telecommunications Conference*, pp. 431-435, 1978.

[12] S.T. Sheu and J. Chen, "A novel delay-oriented shortest path routing protocol for mobile ad hoc networks," in *Proc. of IEEE International Conference on Communications (ICC)*, Helsinki, Finland, pp. 1930-1934, June 2001.

[13] C.E. Perkins, E. Belding-Royer, and S.R. Das, "Ad hoc On-demand Distance Vector (AODV) routing," *IETF RFC* 3561, July 2003.

[14] D.B. Johnson, D.A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile

ad hoc networks (DSR)," *IETF Internet Draft* (work in progress), July 2004.

[15] X. Hong, K. Xu, and M. Gerla, "Scalable routing protocols for mobile ad hoc networks," *IEEE Network*, vol. 16, issue 4, pp. 28-39, July/Aug. 2002.

[16] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, issue 6, pp. 30-39, Nov./Dec. 2001.

[17] G.G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," Technical Report ISI/RR-87-180, Inst. for Scientific Information, Mar. 1987.

[18] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. of Canadian Conference on Computational Geometry*, Vancouver, BC, Aug. 1999.

[19] T.-C. Hou and V.O.K. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, vol. 34, no. 1, pp. 38-44, Jan. 1986.

[20] Trimble Navigation Limited. *Datasheet & specifications for Trimble Thunderbolt GPS Disciplined Clock.* Sunnyvale, CA. Available at http://trl.trimble.com/ docushare/ dsweb/Get/Document-10015/

[21] J. Li, C. Blake, S.J. Douglas, D. Couto, H. Lee, and R. Morris, "Capacity of ad hoc wireless network," in *Proc. of ACM MobiCom*, Rome, Italy, pp. 61-69, Sept. 2001.

[22] S. Roy and J.J. Garcia-Luna-Aceves, "Node-centric hybrid routing for ad hoc wireless extensions of the Internet," in *Proc. of IEEE Globecom*, Taipei, Taiwan, pp. 183-187, Nov. 2002.

[23] S.Y. Ni, Y.C. Tseng, Y.S. Chan, and J.P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proc. of ACM MobiCom*, Seattle, WA, pp. 151-162, Aug. 1999.

[24] Y. Yi, M. Gerla, and T.-J. Kwon, "The selective intermediate nodes scheme for ad hoc on-demand routing protocols," in *Proc. of IEEE International Conference on Communications (ICC)*, New York, NY, pp. 3191-3196, Apr./May 2002.

[25] Z.J. Haas, J.Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. of IEEE Infocom*, New York, NY, pp. 1707-1716, June 2002.

[26] P. Ning and K. Sun, "How to misuse AODV: a case study of insider attacks against mobile ad hoc routing protocols," in *Proc. of IEEE Information Assurance Workshop*, West Point, NY, June 2003.

[27] W. Wang, Y. Lu, and B. Bhargava, "On vulnerability and protection of ad hoc on-demand distance vector protocol," in *Proc. of the International Conference on Telecommunication*, Papeete, France, Feb./Mar. 2003.

[28] J.-H. Song, W.S.V. Wong, V.C.M. Leung, and Y. Kawamoto, "Secure routing with tamper resistant module for mobile ad hoc networks," *ACM Mobile Computing and*

*Communications Review*, vol. 7, issue 3, July 2003.

[29] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. of ACM MobiCom*, Boston, MA, Aug. 2000.

[30] The network simulator - NS-2 notes and documentation and source code. Available at http://www.isi.edu/nsnam/ns/

[31] NS-2 for grid. Available at http://www.pdos.lcs.mit.edu/grid/sim/index.html

# Chapter 2 Efficient On-Demand Routing for Mobile Ad hoc Wireless Access Networks[1]

## 2.1 Introduction

In this chapter, we present an efficient on-demand routing mechanism, which can find a route to the gateway with a controlled amount of routing overhead [1][2]. We propose a novel extension of the Ad hoc On-demand Distance Vector (AODV) [3] routing protocol for *mobile ad hoc wireless access networks*, which applies the concept of load-balancing to limit the amount of routing control packets. In our proposed scheme, AODV route selection is regulated by a distributed grouping mechanism, which divides the mobile nodes logically into different groups to reduce and distribute routing traffic over the network. Load-balancing (LB) is accomplished by balancing the number of source nodes among the groups, a process that can be controlled and updated by the gateway(s). Simulation results show that as traffic increases, our proposed extension of the AODV routing protocol with Load-Balancing (LB-AODV) has a significantly higher packet delivery fraction, a lower end-to-end delay, and a reduced routing overhead when compared with both AODV and gossip-based routing protocols.

The rest of this chapter is organized as follows. Sections 2.2 give overviews on on-demand routing protocols for MANETs. LB-AODV is described in Section 2.3. Simulation results for performance comparisons are presented in Section 2.4. A summary is given in Section 2.5.

## 2.2 On-Demand Routing Protocols for MANETs

Recently, several reactive (or on-demand) routing protocols for MANETs have been proposed in

[1] Paper published in *IEEE Journal on Selected Areas in Communications*, special issue on *Quality of Service in Variable Topology Networks*, vol. 22, no. 7, pp.1374-1383, Sept. 2004.

the literature. The key motivation behind the design of on-demand protocols is the reduction of the routing load for increasing efficiency. In this section, we provide an overview of these protocols.

### 2.2.1 Ad hoc On-demand Distance Vector (AODV)

The AODV routing protocol [3] is an on-demand variation of the distance vector routing protocol. When a source node desires to send a message to a certain destination node to which it does not have a valid route, it initiates a route discovery process. The source node broadcasts an RREQ (Route REQuest) message to its neighbors, which then forward the request to their neighbors, and so on, until either the destination or an intermediate node with a route to the destination in its routing table is reached. During the process of forwarding the RREQ, an intermediate node record in its routing table (i.e., precursor list) the address of the neighbor from which the first copy of the broadcast packet is received, thereby establishing a reverse path. Additional copies of the same RREQ received later are discarded. Once the RREQ reaches the destination or an intermediate node with a route, the respective node responds by unicasting an RREP (Route REPly) message back to the neighbor from which it first received the RREQ, which relays the RREP backward via the precursor nodes to the source node.

Routes are maintained as follows: when a source node moves, it has to re-initiate the route discovery process to find a new route to the destination. On the other hand, when an intermediate node along the route moves, its upstream neighbor will notice route breakage due to the movement and propagate an RERR (Route ERRor) message to each of its active upstream neighbors. These nodes in turn propagate the RERR packet to their upstream neighbors, and so on until the source node is reached. The source node may then choose to re-initiate the route discovery for that destination if a route is still desired.

### 2.2.2 Dynamic Source Routing (DSR)

Another well studied on-demand routing protocol is the Dynamic Source Routing (DSR) [4] protocol. DSR uses source routing rather than hop-by-hop routing, with each packet to be routed carrying in its header the complete, ordered list of nodes through which the packet must pass. These routes are stored in a route cache. The key advantage of source routing is that intermediate nodes do not need to maintain up-to-date routing information in order to route the packets they forward, since the packets themselves have already included the path information. This fact, coupled with the on-demand nature of the protocol, eliminates the need for periodic route advertisement present in other protocols.

### 2.2.3 Gossip-based Routing

The two on-demand routing protocols described in the previous sub-sections are based on some variant of flooding. Despite various optimizations such as an expanding ring search technique [3] and selective intermediate nodes scheme [5], many routing messages may be propagated unnecessarily. Gossip-based routing [6] aims to reduce the routing control overhead by discarding some routing messages with certain probabilities.

The basic idea of the gossip-based routing protocol [i.e., GOSSIP1($p$)] is as follows: A source node sends the RREQ with probability 1. When an intermediate node first receives an RREQ, it will either broadcast the RREQ to its neighbors with probability $p$ or discard it with probability $1 - p$. If the node receives the same RREQ again, the packet will be discarded. Therefore, all mobile nodes broadcast a given RREQ at most once.

If the source node has relatively few neighbors, there is a chance that none of these neighbors will broadcast the RREQ. Thus, a route to the destination may not be obtained. To prevent this from happening, it is recommended the use of probability 1 for the first $k$ hops before continuing with probability $p$ for subsequent hops [i.e., GOSSIP1($p$, $k$)].

## 2.3 Ad hoc On-Demand Distance Vector Routing Protocol with Load-

## Balancing (LB-AODV)

In this section, we begin by describing the rationale and operation of our load-balancing mechanism based on grouping of mobile nodes. The operation of the proposed LB-AODV routing protocol is explained in the Sections 2.3.2 and 2.3.3. It is followed by a discussion on the selection of the total number of groups in Section 2.3.4. The balance index update procedures are introduced in Section 2.3.5. Finally, we compare the route discovery processes among AODV, gossip-based, and LB-AODV routing protocols in Section 2.3.6.

We shall initially consider a mobile ad hoc wireless access network with a single gateway. The following terminologies will be used in this study: A *source node* is a mobile node with data packets to send towards the gateway. A *common node* is a mobile node that does not have data to send and does not belong to any particular group. An *active node* is a mobile node that has valid route(s) to the gateway and is currently being used to forward packets towards the gateway.

### 2.3.1 Load-balancing Mechanism with Grouping

We propose a load-balancing mechanism based on the concept of grouping. It reduces the number of unnecessary retransmissions of routing messages and prevents network congestion by separating source nodes into different *groups* and allowing source nodes to relay only packets generated by their own group members and common nodes [1][2].

The basic idea of our grouping mechanism is to partition all mobile nodes into several logical divisions such as $A$, $B$, $C$, $D$, and $E$ as shown in the example in Figure 2.1. All common nodes belong to the division $E$ in this example, and they are allowed to relay packets from any groups towards the gateway. On the other hand, a source node, which belongs to one of the groups $A$, $B$, $C$, and $D$ in this example, is not allowed to relay packets from other than its own group. For example, packets generated by any members of group $A$ can be relayed only by other

source nodes of group $A$ and common nodes belonging to division $E$.

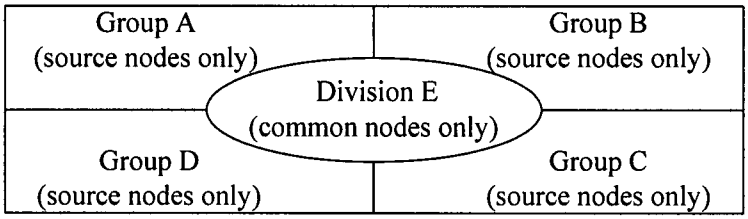



Figure 2.1 An example of logical partitioning of mobile nodes.

In the route discovery process, an RREQ message is only forwarded to the common nodes and those nodes that belong to the same group. Thus, the amount of control traffic can be reduced. The determination of the number of groups is an important consideration in the operation of the LB-AODV routing protocol, and is discussed in detail in Section 2.3.4.

By dividing source nodes into several groups, the packet relaying responsibility and the traffic load can be distributed among different groups. The proposed load-balancing mechanism aims at maximizing the balance index $B$, which is defined as [7]:

$$B = \frac{\left[\sum_{i=1}^{G} f_i\right]^2}{G\sum_{i=1}^{G} f_i^2} \tag{2.1}$$

where $f_i$ denotes the total number of source nodes of group $i$, and $G$ denotes the total number of groups. Given the number of groups $G$, the balance index converges to one when the total number of source nodes of each group approaches equality, while it approaches $1/G$ when all source nodes of the network are assigned to the same group. In our LB-AODV routing protocol, the *state information* is a $(G + 1)$ - tuple in the form of <*group number*, $f_1$, $f_2$, ..., $f_G$>. This information is maintained at all active mobile nodes.

The idea of grouping nodes in LB-AODV is similar to the concept of *routing zone* in

the Zone Routing Protocol (ZRP) [8]. Both routing protocols send queries only to selected nodes in the network during the route discovery process. However, the design goals for these two protocols are different. ZRP targets towards self-organized mobile ad hoc networks while LB-AODV targets towards mobile ad hoc wireless access networks in which mobile nodes can access the Internet via stationary gateway node(s). In addition, the zone partitioning in ZRP is physical in which nodes within certain number of hops are being grouped together. On the other hand, LB-AODV partitions mobile nodes into several logical divisions in order to maximize the balance index. Furthermore, ZRP belongs to a family of hybrid proactive/on-demand routing protocols, whereas LB-AODV is a purely on-demand routing protocol.

### 2.3.2 Load-balancing Route Decision Process

Using the load-balancing route discovery process, we can dynamically minimize the variance of the total number of source nodes between groups. The flow charts for the route selection process are shown in Figure 2.2. A group number is assigned to each source node that initiates the route discovery process.

When a node has data to send but does not know a route to the gateway, the new source node initiates the route discovery process by broadcasting an RREQ message to its neighboring nodes [see Figure 2.2(a)]. When an intermediate node receives the RREQ packet, it processes this message according to its state information. An intermediate node that is not an active node will simply broadcast this RREQ message to its neighbors. On the other hand, if the intermediate node is an active node, it will calculate the balance index $B$ based on the state information stored in its cache.

If the balance index $B$ can be maximized by accepting this new source node into one of its serving groups, then this intermediate node will send an RREP message to the source node. This RREP message includes information about which group this particular source node has been assigned to. The flow chart for the operations of an intermediate node is shown in Figure 2.2(b).

Since the active intermediate node can assign different groups to the source node according to its state information, it needs to maintain different route entries to the gateway for different groups it is currently serving.

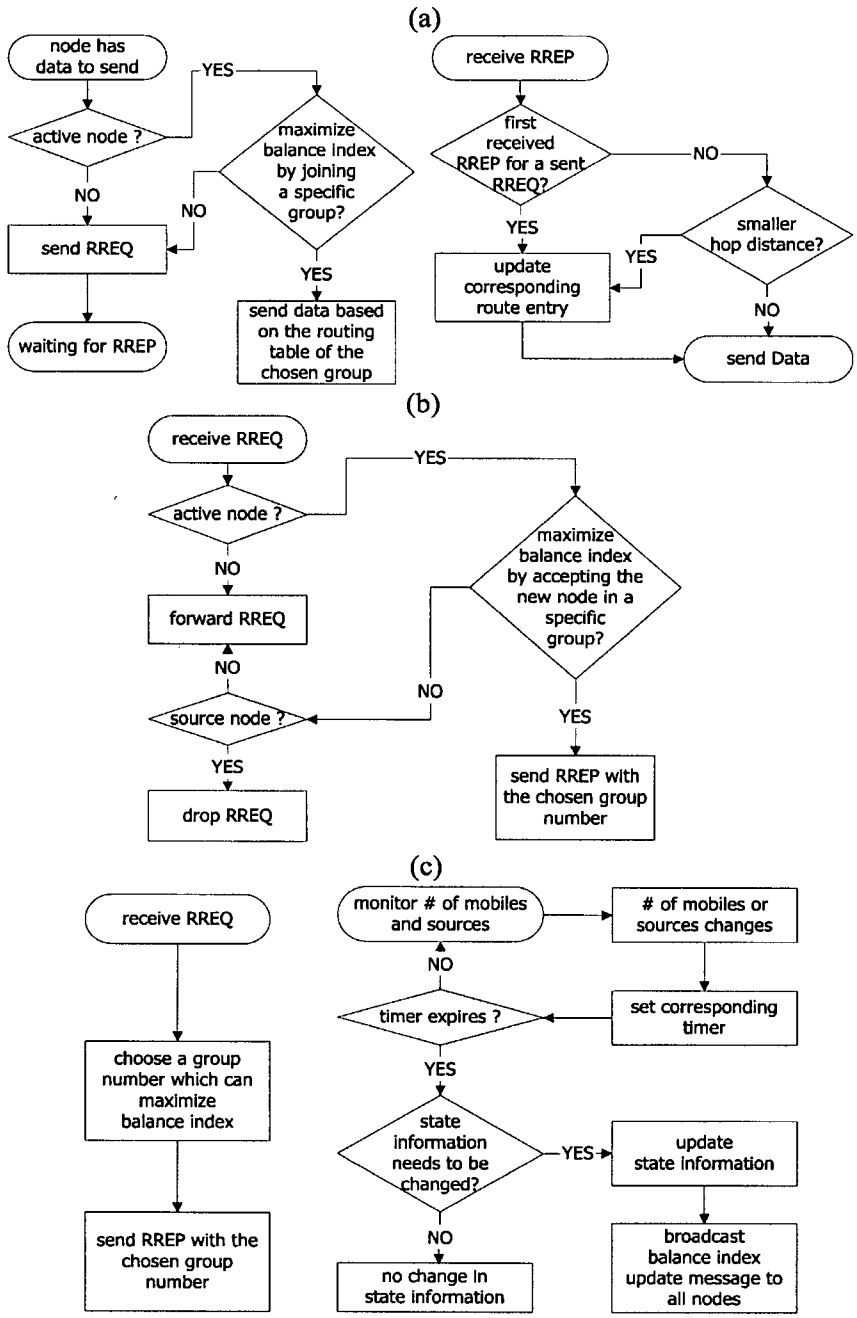



Figure 2.2 Processes in mobile nodes and gateway: (a) source node, (b) intermediate node, (c) gateway node.

Similarly, when the gateway node receives an RREQ message, it will assign a group number to the new source node. The group number is chosen such that the balance index $B$ is maximized. The gateway then sends an RREP message to the source node. The flow chart for the operations of the gateway node is shown in Figure 2.2(c). When the source node receives the RREP message, it will begin sending data packets to the gateway immediately via the node from which it received the RREP message.

Due to transient balance index mismatch, it is possible that a source node may receive multiple RREP packets from several nodes with different group numbers. In this case, the source node will compare the hop-count field in those RREP packets and select the one with the smallest hop-count field value. The group number in the chosen RREP packet will then be used.

### 2.3.3 Load-balancing Route Maintenance Process

When a source node detects a link breakage via an RERR message, it will re-initiate the route discovery by sending an RREQ message with its group number towards the gateway. Those intermediate source nodes that do not belong to this particular group will simply drop the RREQ message. When either an active node (which has a routing cache for this group) or another source node (which belongs to the same group) receives the RREQ message, it will send an RREP message to the source node. The above procedures limit the amount of routing overhead. Note that the balance index remains unchanged after the route discovery process. This is because the new route is still part of the original group.

Due to the topology changes brought about by node mobility, it is possible that the RREQ message may not reach the gateway via the routes in a particular group. We resolve this issue as follows: If the source node has not received any RREP message after a certain period of time, it will re-initiate the route discovery process, as if it was a new source node, by sending another RREQ message without the group number. As long as there exists a route to the gateway, the source node will eventually join another group.

When an active intermediate node becomes a new source node, it first checks the state information stored in its cache. If the balance index can be maximized via one of its serving groups (e.g., group $x$), then this new source node will send data packets to the gateway using group number $x$. Otherwise, this new source node will initiate the route discovery process by broadcasting the RREQ message to its neighboring nodes to find a route that can maximize the balance index.

We assume that soft state information is maintained in the routing cache in each active node. That is, each routing entry has an associated timer. When an intermediate active node or gateway has not received data packets corresponding to a particular entry for a certain period of time, that routing entry and its group number will be deleted.

### 2.3.4 Determining the Number of Groups

The determination of the number of groups is critical for the efficiency of the LB-AODV routing protocol. The number of groups is chosen as a trade-off between the network connectivity and the amount of routing control overhead. To determine the number of groups, the gateway has to obtain the following information: the number of source nodes, the number of mobile nodes, and the size of network. Different methods exist for the estimation of these parameters. In this study, we assume that the gateway can estimate the number of source nodes by monitoring the source address field in the packet header from the packets it received. Assuming that network authorization and authentication are required for the mobile nodes to communicate with the gateway, the gateway can estimate the total number of mobile nodes. If the size of network is not known in advance, the gateway has to estimate the size of the network based on the hop-count information from the packets it received. Due to estimation errors, the size of network and the corresponding group number may not always be correct. To measure the percentage change of throughput due to estimation error, we perform a sensitivity analysis in Section 2.4.3.

It has been shown [9] that for normal MANET scenarios, the best performance can be

achieved when the average number of neighbors is around seven. In this study, we define the *optimal number of mobile nodes R* in a MANET *topology* as the number of mobile nodes that results in the average number of neighbors being around seven. If the gateway can estimate the size of the network correctly, it can calculate the value of $R$ by assuming that all the nodes are uniformly distributed in the network.

We aim to minimize the difference between the optimal number of mobile nodes $R$ and the variable $T$, which is defined as *the total number of mobile nodes that can relay packets generated by each group*. The rationale is that the optimal number $R$ gives the best performance without decreasing the network connectivity in a given network.

Given the number of source nodes $S$ and the number of mobile nodes $M$, the total number of common nodes is equal to $M - S$. Given the total number of groups $G$ and assuming that each group has the same number of source nodes, the number of source nodes that belong to each group is $S/G$. The total number of mobile nodes $T$ that relay packets generated by each group is given by $M - S + (S/G)$. Therefore, the gateway chooses the number of groups $G$ such that the absolute difference between $T$ and $R$ is minimized:

$$\begin{aligned} G &= \underset{g \in \{1,2,\cdots,S\}}{\arg\min} \left| T - R \right| \\ &= \underset{g \in \{1,2,\cdots,S\}}{\arg\min} \left| M - S + (S/g) - R \right| \end{aligned} \tag{2.2}$$

The number of groups $G$ is the function of the value of $M$, $S$, and $R$ as shown in equation (2.2).

1. If $M \geq R + S$, then $G = S$. Because the number of common nodes, $M - S$, is greater than or equal to the optimal number of mobile nodes $R$ in a given topology, a single source node per group will minimize $\left| T - R \right|$.
2. If $R < M < R + S$, then $G$ is equal to one of $\lfloor S/(R+S-M) \rfloor$ or $\lceil S/(R+S-M) \rceil$ which

minimizes $|T-R|$.

3. If $M \leq R$, then $G = 1$. Since the total number of mobile nodes $M$ is less than the minimum required number of mobile nodes $R$, all the mobile nodes have to join the same group to minimize $|T-R|$. In this case, our LB-AODV routing protocol is identical to the original AODV.

Consider the following example: The network topology is 1500×300 ($m^2$). The location of all the nodes is uniformly distributed in the network. The module *setdest* in network simulator (*ns*2) [10] can be used to calculate $R$ such that the average number of neighbors is around 7. Based on this calculation, $R$ is equal to 30. In this case, if the number of source nodes $S$ is 25 and the number of mobile nodes $M$ is 50, then from equation (2.2), the number of groups $G$ is equal to 5.

### 2.3.5 Balance Index Update

Our proposed LB-AODV can also support dynamic changes in the number of groups as the number of mobile nodes changes due to either join or leave operations. We assume that the gateway monitors the total number of mobile nodes $M$ and the number of source nodes $S$ periodically. Whenever the optimal number of groups $G$ or the number of source nodes has changed [from equation (2.2)], the state information needs to be updated based on equation (2.1). The gateway then broadcasts an advertisement message to all the nodes to update the state information. The update information includes: (1) the number of source nodes in each group $< f_1, f_2, \ldots, f_G >$; and (2) the addresses of those source nodes that have been reassigned to different groups and their newly assigned group numbers. For those source nodes that have been assigned new group numbers, they will re-initiate the route discovery process again by including the new group number in the subsequent RREQ messages.

### 2.3.6 Comparison of Route Discovery Processes

In this subsection, we describe the differences in the route discovery procedures among the AODV, gossip-based, LB-AODV routing protocols. Consider a MANET with a large number of mobile nodes. We assume that each mobile node has $n$ neighbors within its transmission range, and none of the mobile nodes has a route entry to the requested destination. Suppose a source node $S$ sends an RREQ message to its neighboring nodes. In all three routing protocols, we consider that when a neighboring node first receives an RREQ packet, the node will either broadcast the RREQ packet to its neighbors with probability $p$ or discard it with probability $1 - p$, where $0 \leq p \leq 1$. If the node receives the same RREQ packet again, the packet will be discarded. Different protocols differ in the possible values of $p$.

In the AODV routing protocol, when a mobile node first receives an RREQ packet and does not have a route entry to the requested destination, it will always broadcast the RREQ packet to its neighbors. Therefore, an RREQ packet will be broadcasted over more than one hop with probability $p = 1$. However, the number of RREQ packets being broadcasted is proportional to the number of nodes, and cannot be controlled or regulated.

Consider the basic gossip-based routing protocol (e.g., GOSSIP1($p$) in [6]). When a mobile node first receives an RREQ packet, it will either broadcast the RREQ packet to its neighbors with probability $p$ or discard it with probability $1 - p$. Therefore, an RREQ packet will be broadcasted over more than one hop with probability $1 - (1 - p)^n$ where $n$ is the number of neighbors. In our simulation model, we use the modified GOSSIP1($p$, $k$). In GOSSIP1($p$, $k$), when a mobile node first receives an RREQ packet, with probability 1 it will broadcast the RREQ packet to its neighbors for the first $k$ hops. However, after $k$ hops from the source node $S$, GOSSIP1($p$, $k$) works exactly the same way as GOSSIP1($p$). Other variations of the gossip-based routing protocols have been proposed recently (e.g., [11][12]). Performance comparisons between LB-AODV and these protocols are subject of future work.

Consider the LB-AODV routing protocol. When a source node sends an RREQ message, $m$ out of $n$ neighboring nodes will broadcast the RREQ packet to its neighbors while the other neighboring nodes will discard the packet. Therefore, an RREQ packet can be broadcasted over more than one hop with probability $p = 1$ if the group number is chosen correctly. Since the LB-AODV routing protocol regulates the number $m$ dynamically, it can control the number of RREQ packets being broadcasted without degrading the level of network connectivity.

## 2.4 Simulation Model and Evaluations

In this section, we compare the performance between our proposed LB-AODV [1][2], the original AODV [3], and the gossip-based routing [GOSSIP1($p$, 1)] [6] protocols.

### 2.4.1 Simulation Model

The Network Simulator (*ns2*) [10] is used for the implementation of LB-AODV and GOSSIP1 routing protocols. The physical radio characteristics of each mobile node's radio interface are chosen to approximate the Lucent WaveLAN [13] operating as a shared-medium radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 $m$. For the medium access control layer, the IEEE 802.11 Distributed Coordinated Function (DCF) [14] is used. The propagation model combines both a free space propagation model and a two-ray ground reflection model. We use the same configuration parameters as those of ns-2 version b8a.

Constant Bit Rate (CBR) traffic sources are used with different packet generation rates. The data packet size is 512 bytes. For the simulation results presented in Figures 2.3–2.13, the size of the network is 1500×300 ($m^2$) and the number of mobile nodes is 50. On the other hand, for the simulation results shown in Figures 2.14–2.15, a 1500×600 ($m^2$) topology is used with 100 mobile nodes. Finally, for the results presented in Figures 2.16–2.17, a 1000×1000 ($m^2$) topology is used with various number of mobile nodes. Table 2.1 provides a summary of the

simulation parameters. One stationary gateway node is located in the middle of the grid [i.e., coordinate (750, 150)] for the first three simulation scenarios. A *random waypoint model* [15] is used for the mobility model. Each node moves at a speed that is uniformly distributed from 0 to 20 m/s. Each simulation run takes 900 simulated sec. The results presented are mean values of at least 10 simulation runs; the error bars represent the 95 % confidence intervals about the means in Figures 2.3–2.15. For fair comparisons, all three routing protocols use the same set of mobility and traffic scenarios.

Table 2.1 Simulation parameters.

| | | | |
|---|---|---|---|
| Transmission range | 250 m | Topology size | 1500×300 $m^2$<br>1500×600 $m^2$<br>1000×1000 $m^2$ |
| Bandwidth of radio interface | 2 Mb/sec | Traffic type | CBR |
| Simulation time | 900 sec | Packet generation rate | 3, 4, 6, 8, 9 packets/sec |
| Number of nodes | 27, 47, 50, 66, 84, 100 | Packet size | 512 Bytes |
| Number of source nodes | 10, 20, 25, 30, 40 | Pause time (second) | 100, 300, 500, 600, 700, 900 |

For comparisons with gossip-based routing, since only $T$ mobile nodes can relay packets generated by each group in LB-AODV, we choose the gossip probability $p$ to be equal to $T/M$. Thus, after $k$ hops from the source node, when a neighboring node first receives an RREQ packet, it will either broadcast the RREQ packet to its neighbors with probability $T/M$, or discard it with probability $1 - T/M$. Table 2.2 provides a summary of the values of $T$, $G$, and $p$ by varying the number of source nodes $S$. The value of $k$ is chosen as 1 because the average path length is about 2.5.

### 2.4.2 Performance Metrics

Table 2.2 Simulation variables.

| *Variables* / *Number of sources, S* | *Number of possible relay nodes, T* | *Number of groups, G* | *Gossip probability, p* |
|---|---|---|---|
| 10 | 41 | 10 | 41/50 |
| 20 | 31 | 20 | 31/50 |
| 25 | 30 | 5 | 30/50 |
| 30 | 30 | 3 | 30/50 |
| 40 | 30 | 2 | 30/50 |

The following performance metrics are used for comparisons. The *packet delivery fraction* is defined as the measured ratio of the number of data packets delivered to the destinations to the number of packets generated by all traffic sources. The *average end-to-end delay of transferred data packets* includes all possible delays caused by buffering during route discovery, queuing at the interface-queue, retransmission delays at the medium access control layer, propagation and transmission times. The *normalized control overhead* is defined as the number of both routing and update (in LB-AODV) packets transmitted per data packet delivered at the destination. Note that each time a packet is forwarded is counted as one packet transmission.

### 2.4.3 Performance Comparisons

Scenario 1: Single Gateway, Multiple Source Nodes with Same Packet Generation Rate

Figures 2.3–2.5 show the performance of the network with different number of CBR sources. When the number of sources is less than 20, all three routing protocols provide a high packet delivery fraction, small end-to-end delay and normalized control overhead. Results in Figure 2.3 indicate that LB-AODV improves the packet delivery fraction by 15 % over the other schemes when the number of sources increases to 25. As traffic further increases, the improvement is increased radically. This implies that when traffic load is high (i.e., more than 25 sources in this scenario), most of the routes towards the gateway are congested by many control and data

packets. Therefore, contention and collision between neighbors increase exponentially, and thus the AODV and GOSSIP1 routing schemes become less efficient. Results in Figure 2.4 indicate that within a given end-to-end delay constraint, LB-AODV can support more traffic when compared with the other protocols. Figure 2.5 shows that LB-AODV has a much lower normalized control overhead when compared with AODV and GOSSIP1 routing protocols.

Figures 2.6–2.8 show the overall performance by varying pause time (i.e., mobility). The number of source nodes is equal to 25. These results indicate that in a slightly congested network (with 25 source nodes) LB-AODV maintains a better performance over different mobility rates when compared with AODV and GOSSIP1.

Scenario 2: Single Gateway, Multiple Source Nodes with Different Packet Generation Rates

Our proposed load-balancing mechanism distributes the number of source nodes evenly among different groups. Therefore, it cannot balance the average packet transmission rates of each group if each source node has a different packet generation rate. In this simulation, we investigate the effects of source nodes with different packet generation rates on the performance of LB-AODV routing protocol. Table 2.3 provides six cases where source nodes with different packet generation rates of 3, 6 and 9 packets/sec are mixed. The pause time is equal to 500 sec in this scenario. Note that for fair comparisons the average packet generation rate in each scenario is equal to 120 Kbits/sec.

Table 2.3 Transmission scenarios.

| | S1 | S2 | S3 | S4 | S5 | S6 |
|---|---|---|---|---|---|---|
| *Number of source nodes with 3 packets/sec* | 0 | 2 | 4 | 6 | 8 | 10 |
| *Number of source nodes with 6 packets/sec* | 20 | 16 | 12 | 8 | 4 | 0 |
| *Number of source nodes with 9 packets/sec* | 0 | 2 | 4 | 6 | 8 | 10 |

Figures 2.9–2.11 compare the performance of AODV, GOSSIP1, and LB-AODV for the

six scenarios shown in Table 2.3. As shown in all three Figures 2.9–2.11, the performance of LB-AODV is almost constant among different scenarios. Moreover, in all the scenarios considered, LB-AODV consistently and significantly outperforms AODV and GOSSIP1 routing protocols. Results in Figures 2.9–2.11 confirm that the performance gain of LB-AODV over AODV and GOSSIP1 is caused mainly by the reduction of the number of control packet transmissions. Thus, LB-AODV is also efficient in mobile ad hoc wireless access networks that are composed of source nodes with different packet generation rates.

Scenario 3: Single Gateway and Variable Number of Source Nodes

When the number of source node changes, the gateway has to update the value of the balance index and broadcast an advertisement message to all the nodes to update the state information. Therefore, it is expected that as the number of source node changes frequently, a considerable number of control packets be propagated in the network for state information update. In this set of simulations, we investigate the effects of broadcasting of control packet in LB-AODV. The pause time is set to 500 sec, and the maximum number of source nodes is 25. The packet generation rate for each source node is 8 packets per sec. By changing the average session time between communication pairs, we can obtain the results for different traffic densities. Both the average session time and the average inter-session time are assumed to follow the exponential distribution. As described in Section 2.3.3, when the gateway has not received data packets corresponding to a particular entry for a certain period of time (10 sec in this case), the corresponding routing entry and its group number will be deleted.

Figures 2.12–2.13 show the performance by varying the average session time. The average intersession time is set to 60 sec. Figures 2.12–2.13 show that when the average session time is less than 40 sec, AODV works slightly better than LB-AODV with a lower control overhead. On the other hand, LB-AODV outperforms AODV as the average session time increases. We observe that LB-AODV has a lower normalized control overhead as the average

session time increases. Note that as the traffic density increases, with AODV, the network becomes congested with routing and data packets. On the other hand, the grouping mechanism in LB-AODV controls the amount of routing control overhead. It remains efficient even when the number of source nodes changes.

Scenario 4: Two Gateways

This experiment relates to the study of scalability with two gateways. We determine the variation of the throughput (i.e., the total amount of bytes received without errors by the destination per sec) and the normalized control overhead by increasing the network size to a 1500×600 $m^2$ topology and changing the number of mobile nodes to 100. We consider 40 CBR sources, each with a packet generation rate of 4 packets per sec. Since the number of common nodes $M - S$ = 60 exceeds the optimal number $R$ = 50 for this topology, we choose the maximum number of source nodes, 40, as the total number of groups $G$. Therefore, each source node belongs to a different group (refer to Section 2.3.4). The simulation time is 900 sec. All the other simulation parameters remain the same.

In this scenario, two gateways, $G1$ and $G2$, are located in the coordinates (750, 150) and (750, 450), respectively. Since each gateway can monitor the number of source nodes being served, each gateway communicates with 20 source nodes at a maximum. In LB-AODV, the total number of mobile nodes $T$ that relay packets generated by each group is 61 [i.e., $M - S + (S/G)$ = 100 − 40 + (40/40) = 61]. The *state information* should be a $(G + 2)$ - tuple in the form of <*gateway number, group number,* $f_1, f_2, \ldots, f_G$ > in this scenario.

Figure 2.14 shows the throughput as a function of pause time in the network. Since LB-AODV can divide only source nodes into different groups, the increase of control overhead is unavoidable as the number of mobile nodes increases. However, due to the fact that the grouping mechanism can reduce the amount of routing control overhead (see Figure 2.15) and distribute the number of source nodes between two gateways, the throughput of LB-AODV is

approximately three times higher than that of AODV and GOSSIP1 routing protocols. These results show that LB-AODV is still efficient in scenarios with two gateways and a large network size. Note that GOSSIP1 shows a better performance than AODV in this scenario. This is because in large and dense networks gossip-based routing protocols are effective in improving the efficiency by reducing the transmissions of routing control packets [6]. Further performance improvements for LB-AODV may be possible by refining the group assignment algorithm to take into account of the number and location of gateways. This is a subject for further research.

Scenario 5: Sensitivity Analysis

Recall that the optimal group number $G$ is a function of $M$, $S$, and $R$ [see equation (2.2) in Section 2.3.4]. Although the parameters $M$ and $S$ can be monitored by the gateway, the value of $R$ may not always be estimated correctly. If that is the case, the resulting number $G$ may not indeed be optimal in terms of node density. We are interested in determining the percentage change of the throughput as a function of the variations of the size of network, $Z$. The procedures for the sensitivity analysis consist of the following steps:

1. Given the actual size of the network $Z$, we first determine the optimal value of $R$.
2. Given the values $R$, $M$ and $S$, we determine the optimal group number $G$ based on equation (2.2).
3. Given the values $R$, $M$, $S$, and $G$, the expected throughput can be obtained via the *ns*2 simulation. We denote the value as *Throughput* (*optimal*).
4. Let $Z'$ denote the estimated size of network and $\Delta_Z$ denote the percentage change of the size of network. These parameters are related by the following equation:

$$Z' = (1 + \Delta_Z)Z \tag{2.3}$$

   Based on the estimated size of network $Z'$, the sub-optimal value of $R'$ is determined. Similarly, given the values $R'$, $M$ and $S$, the sub-optimal group size $G'$ can be calculated

based on equation (2.2). The sub-optimal expected throughput, denoted as *Throughput* (*sub-optimal*), is obtained via the *ns*2 simulation.

5. The change of the throughput with respect to the variation of the size of network is characterized by the *throughput ratio*, which is defined as: *Throughput* (*sub-optimal*) / *Throughput* (*optimal*).

Figure 2.16 shows the throughput ratio versus the percentage change of the size of network. We assume the actual topology of the network to be 1000×1000 $m^2$. The optimal value of $R$ is 47. There is one stationary gateway located in the coordinate (500, 500). When the size of network is under-estimated by 100 %, the sub-optimal value of $R'$ is 27. On the other hand, the sub-optimal value of $R'$ is 84 when the size of network is over-estimated by 100 %. To study the effect of the number of nodes to the estimated size of network, we vary the number of nodes from 27 to 84. Note that the number 27 and 84 are the sub-optimal values of $R$ when the estimation is deviated by –100 % and +100 %, respectively. Figure 2.17 shows that the number of groups $G$ based on the given values of $M$, $S$, and $R$.

When the number of node $M$ is less than or equal to 47, the throughput ratio is not sensitive to the estimated size of network. On the other hand, as the number of nodes $M$ increases, the throughput ratio is sensitive to both under- and over-estimation of the size of network. An over-estimation of the size of network gives a higher throughput ratio than an under-estimation of the same percentage. These results imply that if there is uncertainty in estimating the size of network, it may be better to underestimate its value in order to reduce the throughput ratio difference.

## 2.5 Summary

With flooding-based on-demand route discovery in mobile ad hoc wireless access networks, many routing messages (i.e., RREQ) are propagated unnecessarily. Moreover, the redundancy of routing information (i.e., RREP and RREQ) processed by the gateway is high in the mobile ad

hoc wireless access network. To reduce the overhead of routing messages, we have proposed an extension of the ad hoc on-demand routing protocol by incorporating the concept of load-balancing in this study. Our proposed LB-AODV protocol is simple and well-suited for the mobile ad hoc wireless access network environment.

We have compared the performance of our proposed LB-AODV protocol with both the original AODV and gossip-based routing protocols in different mobility and traffic scenarios. Simulation results show that LB-AODV delivers more data packets to the gateway and decreases the end-to-end delay of packets delivered by reducing the transmissions of routing control messages by 50 % or more. In scenarios with traffic congestion, LB-AODV significantly outperforms AODV and GOSSIP1 routing protocols. We have compared the performance of the protocols in a scenario with a larger number of mobile nodes accessing two gateways. LB-AODV provides significant advantages over AODV and GOSSIP1 in terms of throughput and routing overhead even in a large network with two gateways. Although we have presented the details of LB-AODV based on the AODV routing protocol, the load-balancing concept developed in this study can generally be applied to other on-demand routing schemes. Moreover, we can further improve our load-balancing scheme by distributing the traffic among mobile hosts according to the traffic load of each path [16] instead of using the hop-count metric.

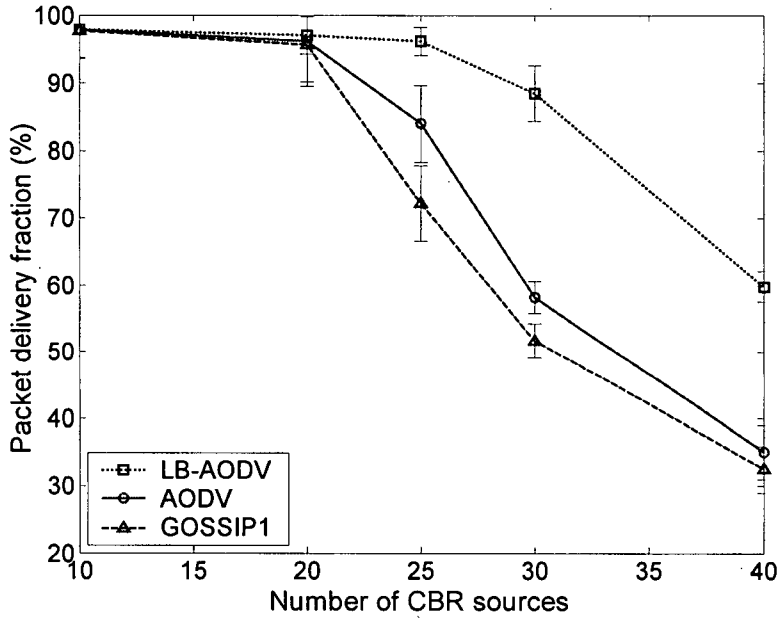



Figure 2.3 Packet delivery fraction among AODV, LB-AODV, and GOSSIP1 routing protocols with varying number of CBR sources (pause time = 500 sec).

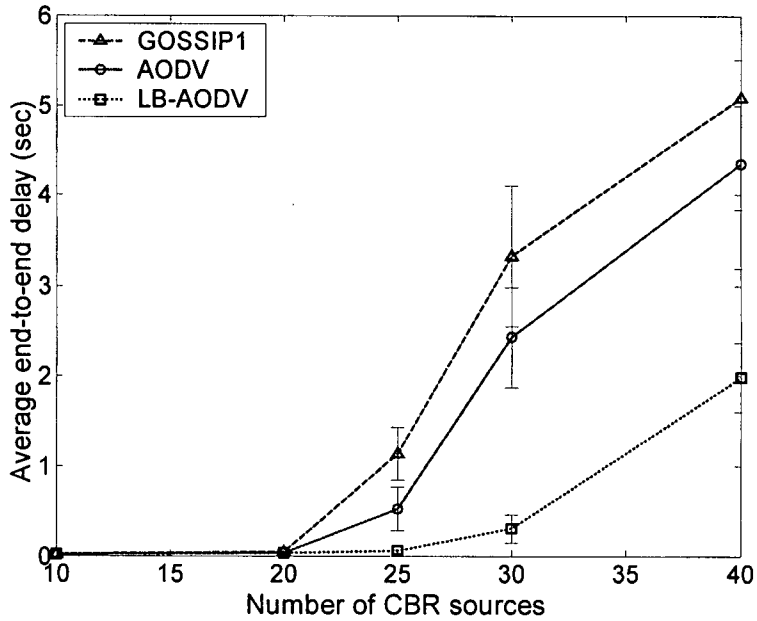



Figure 2.4 Average end-to-end delay among AODV, LB-AODV, and GOSSIP1 routing protocols with varying number of CBR sources (pause time = 500 sec).

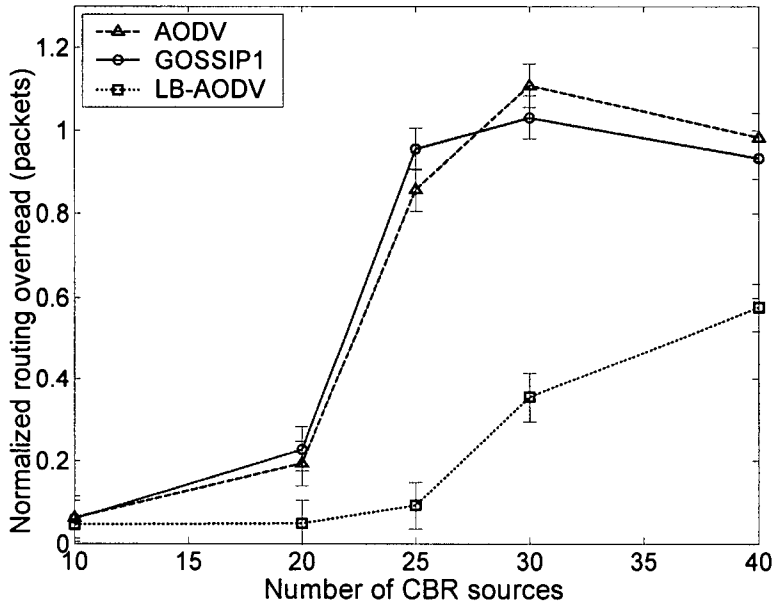



Figure 2.5 Normalized routing overhead among AODV, LB-AODV, and GOSSIP1 routing protocols with varying number of CBR sources (pause time = 500 sec).

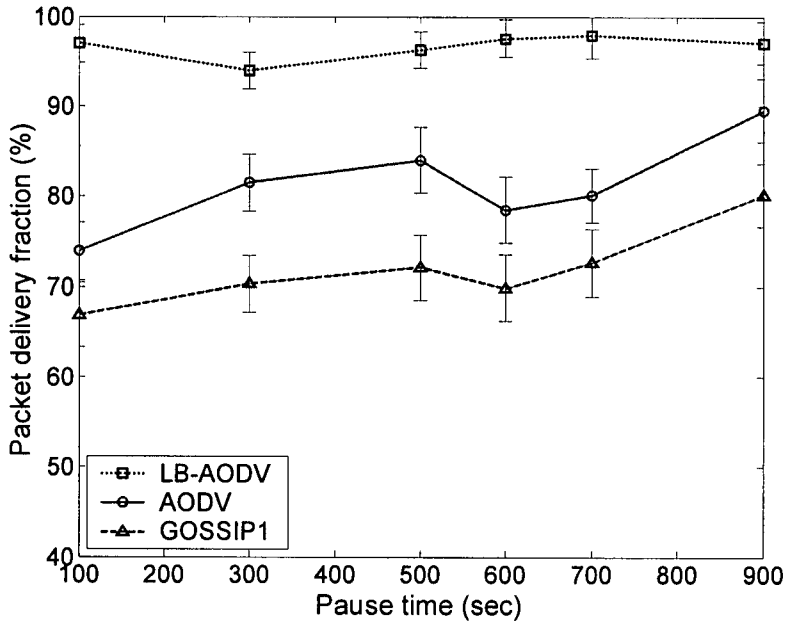



Figure 2.6 Packet delivery fraction among AODV, LB-AODV, and GOSSIP1 routing protocols over a range of pause time (number of CBR sources = 25).

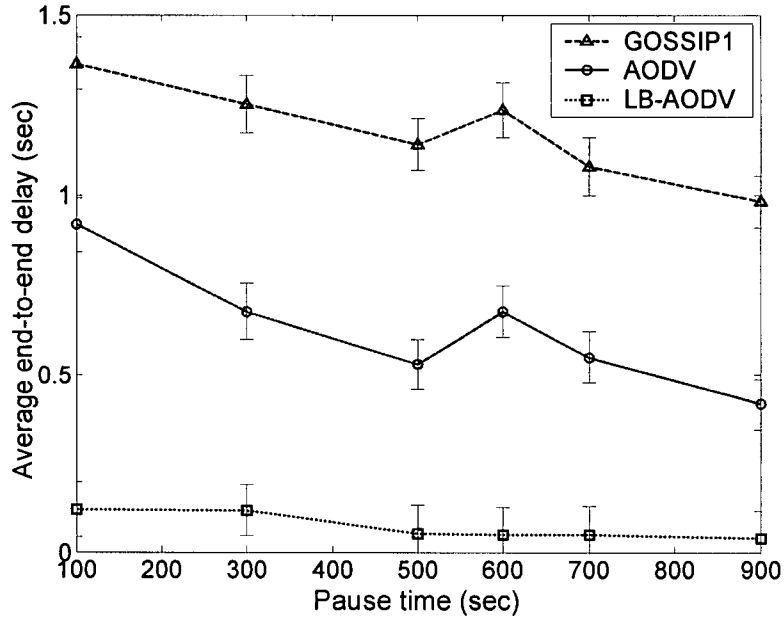



Figure 2.7 Average end-to-end delay among AODV, LB-AODV, and GOSSIP1 routing protocols over a range of pause time (number of CBR sources = 25).

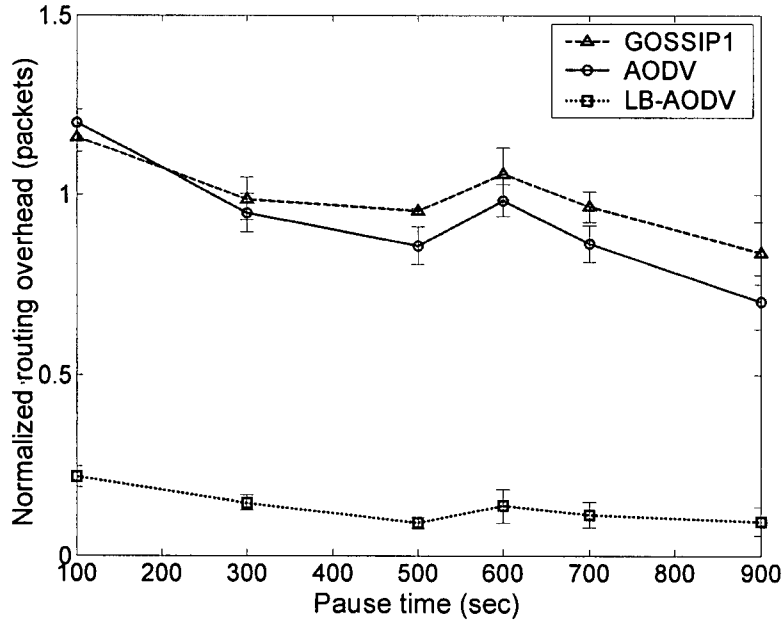



Figure 2.8 Normalized routing overhead among AODV, LB-AODV, and GOSSIP1 routing protocols over a range of pause time (number of CBR sources = 25).

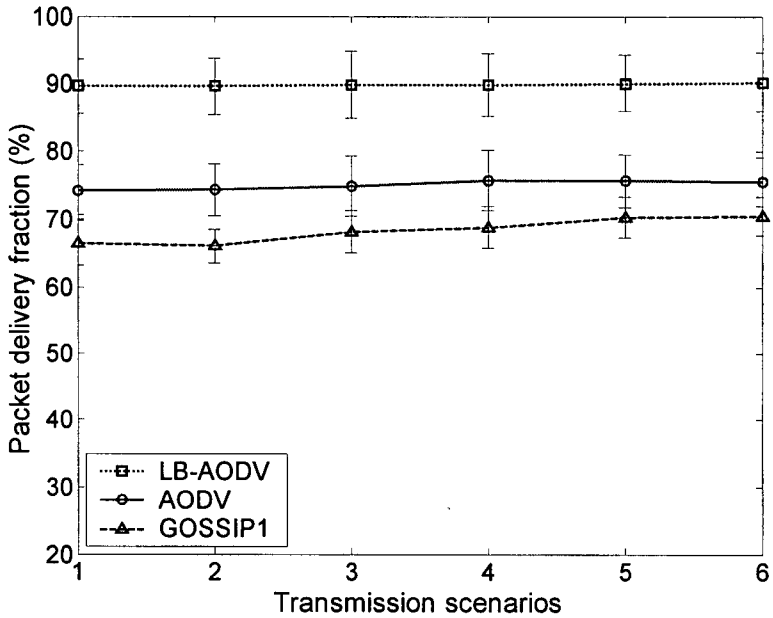



Figure 2.9 Packet delivery fraction among AODV, LB-AODV, and GOSSIP1 routing protocols for variable source rate scenarios shown in Table 2.3 (pause time = 500 sec).

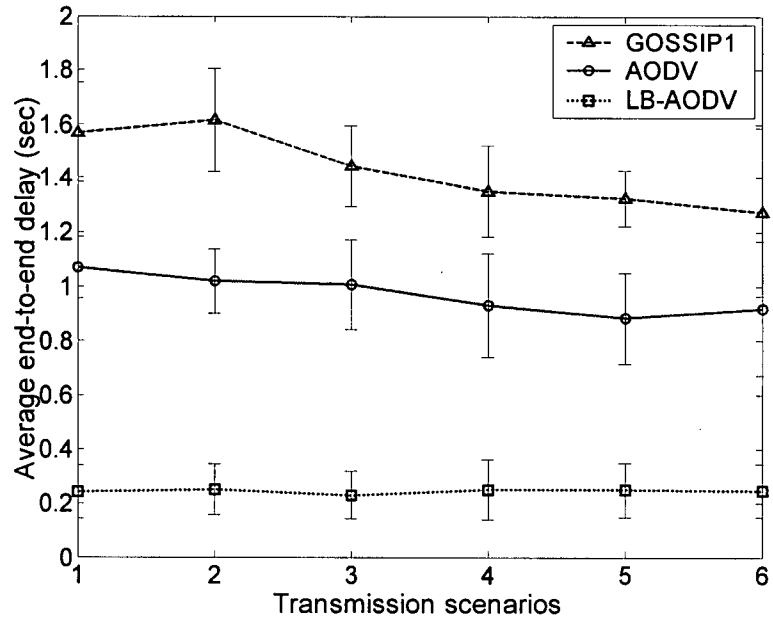



Figure 2.10 Average end-to-end delay among AODV, LB-AODV, and GOSSIP1 routing protocols for variable source rate scenarios shown in Table 2.3 (pause time = 500 sec).

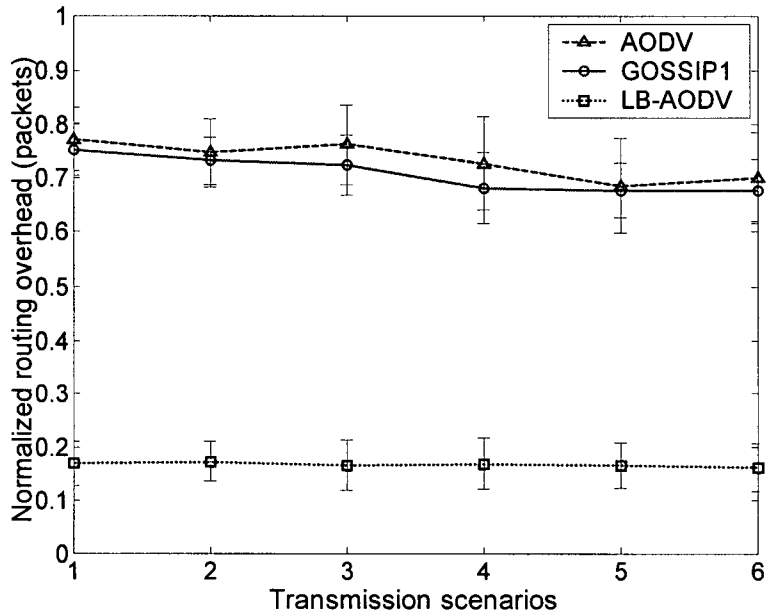



Figure 2.11 Normalized routing overhead among AODV, LB-AODV, and GOSSIP1 routing protocols for variable source rate scenarios shown in Table 2.3 (pause time = 500 sec).

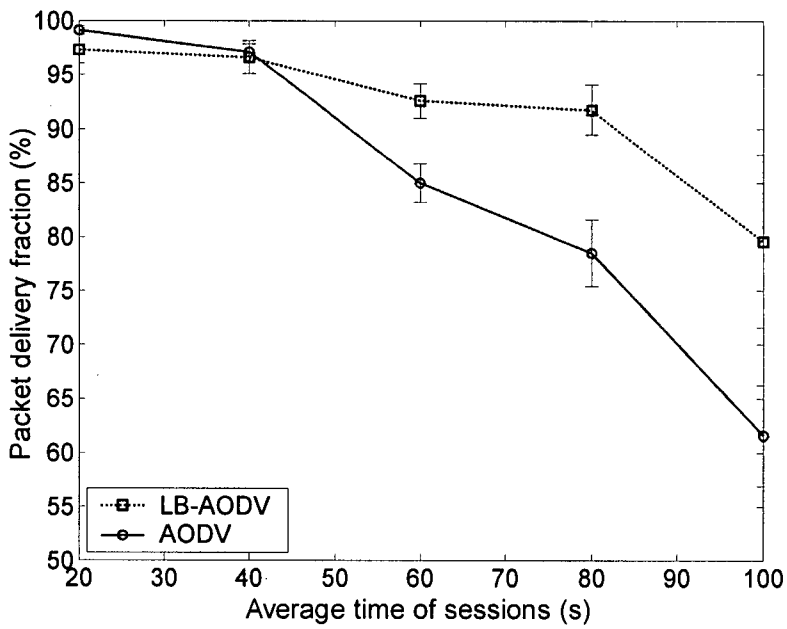



Figure 2.12 Packet delivery fraction among AODV and LB-AODV routing protocols with variable average time of sessions (average intersession time = 60 sec).

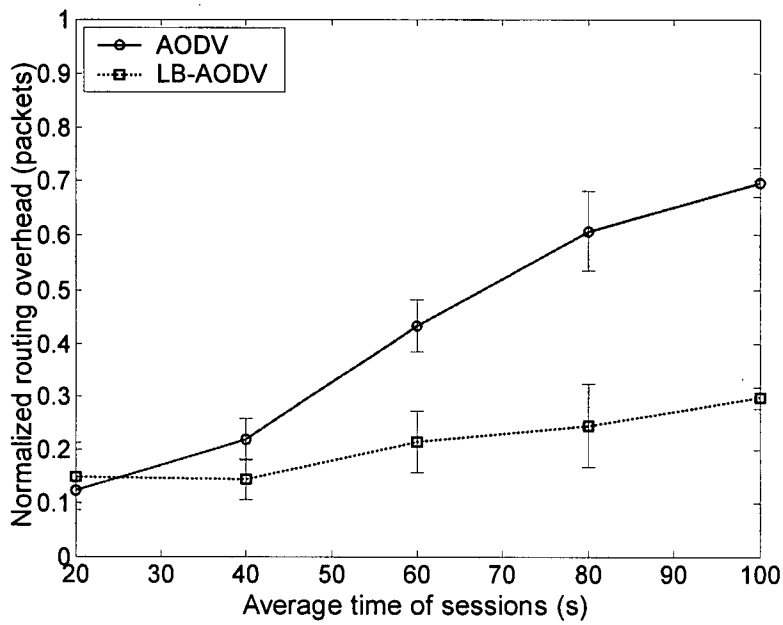



Figure 2.13 Normalized routing overhead among AODV and LB-AODV routing protocols with variable average time of sessions (average intersession time = 60 sec).

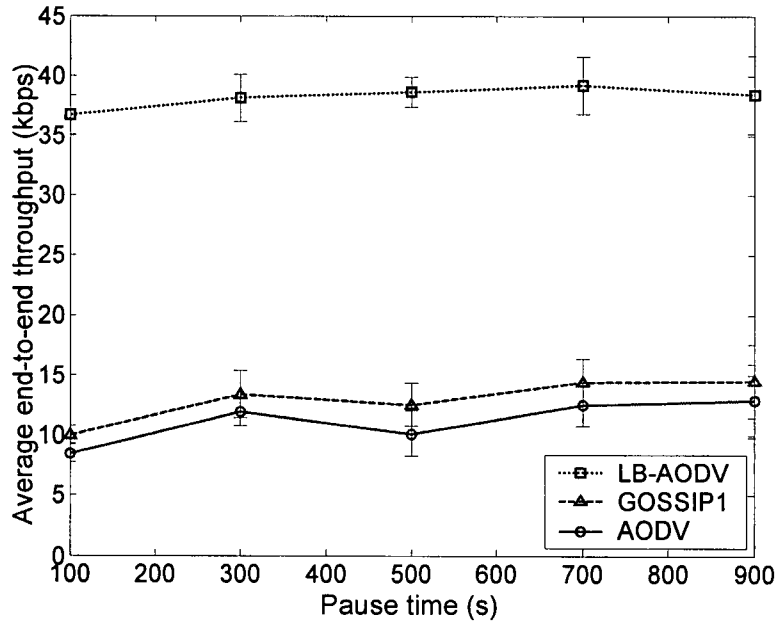



Figure 2.14 Average end-to-end throughput among AODV, LB-AODV, and GOSSIP1 routing protocols in a two-gateway scenario (number of CBR sources = 40).

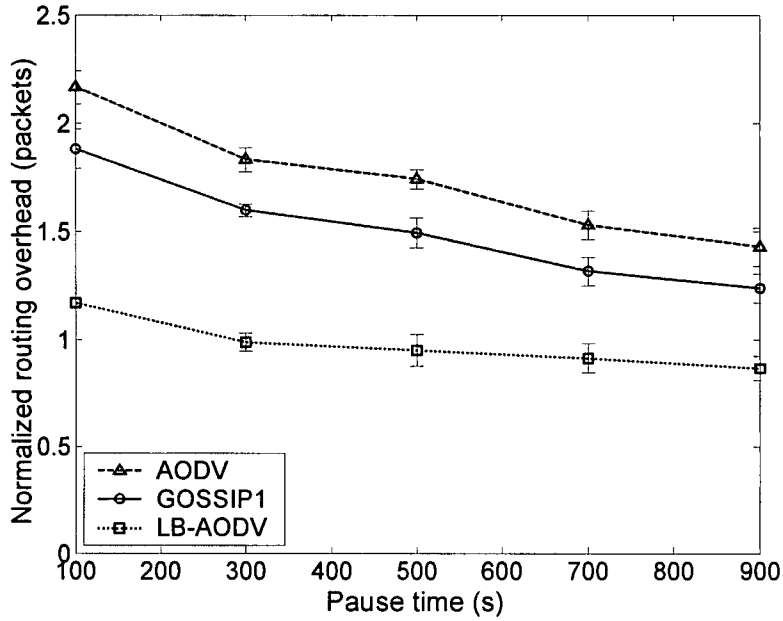



Figure 2.15 Normalized routing overhead among AODV, LB-AODV, and GOSSIP1 routing protocols in a two-gateway scenario (number of CBR sources = 40).

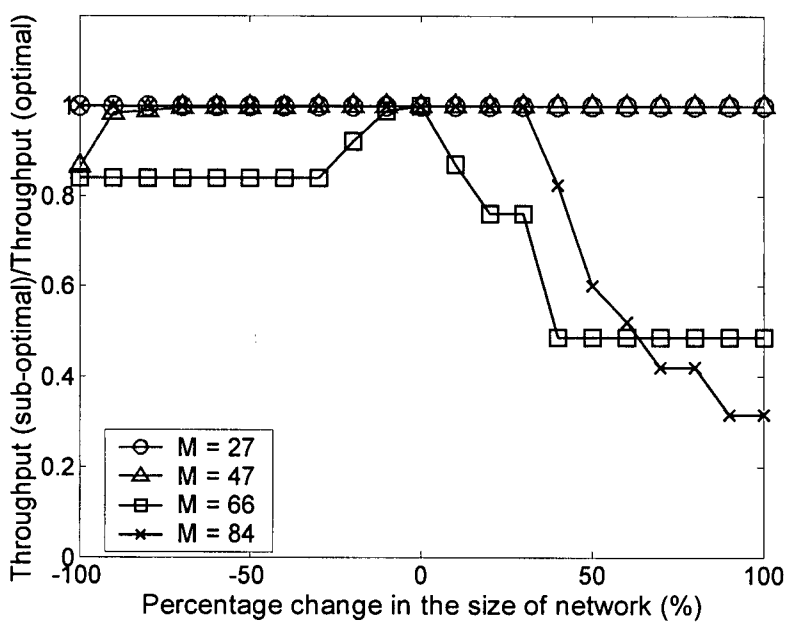



Figure 2.16 Sensitivity analysis of the estimated size of network (number of CBR sources = 25, pause time = 500 sec).

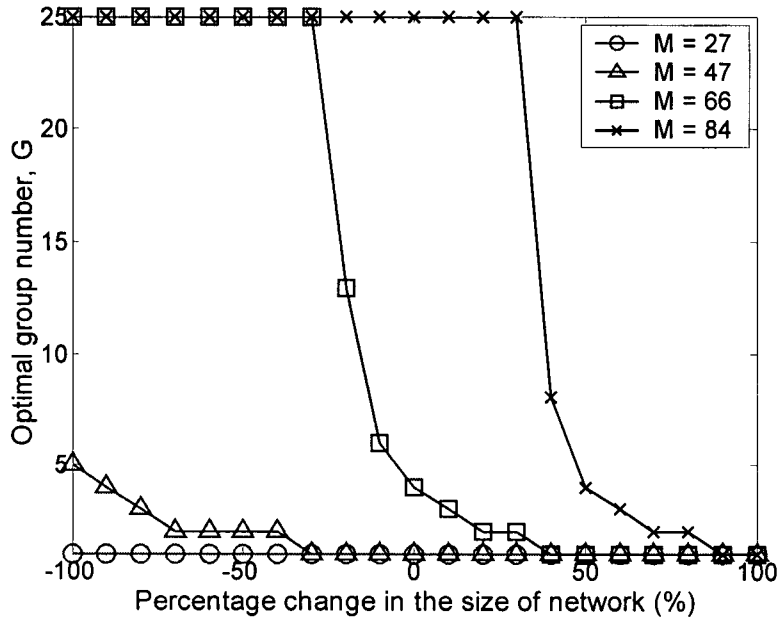



Figure 2.17 Optimal group number of the estimated size of network (number of CBR sources = 25, pause time = 500 sec).

# Bibliography


[1] J.-H. Song, W.S.V. Wong, and V.C.M Leung, "Efficient on-demand routing for mobile ad hoc wireless access networks," in *Proc. of IEEE Globecom*, San Francisco, CA, pp. 558-563, Dec. 2003.

[2] J.-H. Song, W.S.V. Wong, and V.C.M. Leung, "Efficient on-demand routing for mobile ad hoc wireless access networks," *IEEE Journal on Selected Areas in Communications*, vol. 22, no. 7, pp.1374-1383, Sept. 2004.

[3] C.E. Perkins, E. Belding-Royer, and S.R. Das, "Ad hoc On-demand Distance Vector (AODV) routing," *IETF RFC* 3561, July 2003.

[4] D.B. Johnson, D.A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," *IETF Internet Draft* (work in progress), July 2004.

[5] Y. Yi, M. Gerla, and T.-J. Kwon, "The selective intermediate nodes scheme for ad hoc on-demand routing protocols," in *Proc. of IEEE International Conference on Communications (ICC)*, New York, NY, pp. 3191-3196, Apr./May 2002.

[6] Z.J. Haas, J.Y. Halpern, and L. Li, "Gossip-based ad hoc routing," in *Proc. of IEEE Infocom*, New York, NY, pp. 1707-1716, June 2002.

[7] P. Hsiao, A. Hwang, H. Kung, and D. Vlah, "Load-balancing routing for wireless access networks," in *Proc. of IEEE Infocom*, Anchorage, AK, pp. 986-995, April 2001.

[8] M.R. Pearlman and Z.J. Haas, "Determining the optimal configuration for the zone routing protocol," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1395-1414, Aug. 1999.

[9] E.M. Royer, P.M. Melliar-Smith, and L.E. Moser, "An analysis of the optimum node density for ad hoc mobile networks," in *Proc. of IEEE International Conference on Communications (ICC)*, Helsinki, Finland, pp. 857-861, June 2001.

[10] The network simulator - NS-2 notes and documentation and source code. Available at http://www.isi.edu/nsnam/ns/

[11] J. Luo, P.T. Eugster, J.-P. Hubaux, "Route driven gossip: probabilistic reliable multicast in ad hoc networks," in *Proc. of IEEE Infocom*, San Francisco, CA, pp. 2229-2239, Mar./Apr. 2003.

[12] Y. Sasson, D. Cavin, and A. Schiper, "Probabilistic broadcast for flooding in wireless mobile ad hoc networks," in *Proc. of IEEE Wireless Communications and Networking Conference (WCNC)*, New Orleans, Louisiana, pp. 1124-1130, Mar. 2003.

[13] B. Tech, "Development of WaveLAN, an ISM band wireless LAN," *AT&T Technical*

*Journal*, pp. 27-37, July/Aug. 1993.

[14] IEEE Computer Society LAN/MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11," Sept. 1999.

[15] J. Broch, D. Maltz, D. Johnson, Y.-C. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proc. of ACM MobiCom*, Dallas, TX, pp. 85-97, Oct. 1998.

[16] J.-H. Song, W.S.V. Wong, and V.C.M. Leung, "Load-aware on-demand routing (LAOR) protocol for mobile ad hoc networks," in *Proc. of IEEE Vehicular Technology Conference (VTC-Spring)*, Jeju, Korea, pp. 1753-1757, Apr. 2003.

# Chapter 3 Secure AODV Routing Protocol with Table Entry Protection for Mobile Ad Hoc Networks[2]

## 3.1 Introduction

In this chapter, we provide security mechanisms to prevent routing table tampering attacks and to secure both AODV control and data messages in MANETs [1][2]. First, we propose the use of Tamper Resistant Module (TRM) [1] in each mobile node to prevent compromised users from modifying their own routing tables. As an alternative method, we propose the Secure Table Entry Protection (STEP) scheme, which provides the authentication for both the destination sequence number and hop-count fields in the routing table entry. We also propose a variant of STEP, called an Efficient STEP (ESTEP), which can further reduce the control overhead. Second, we propose Secure AODV (SeAODV), a secure routing extension to the original AODV [3] routing protocol. We also propose a Secure Data Forwarding (SDF) scheme based on SeAODV for secure transmissions of data packets over the wireless links. We conduct simulation experiments to determine the performance among STEP, ESTEP, SAODV [4] , SeAODV with SDF, and the original AODV.

The rest of this chapter is organized as follows. Sections 3.2–3.3 give overviews on cryptographic primitives and secure ad hoc routing protocols, respectively. In Section 3.4, we describe the security threats and requirements for AODV routing protocol. We state the assumptions made in our framework and explain their rationales in Section 3.5. In Section 3.6, we describe the use of TRM in mobile ad hoc environments. Our proposed STEP mechanism is described in Section 3.7. In Section 3.8, we describe the operation of SeAODV protocol with SDF mechanism. The performance comparisons among STEP, SAODV, SeAODV with SDF, and

---

[2] Paper submitted to *IEEE Journal on Selected Areas in Communications*, special issue on *Security in Wireless Ad-hoc Networks*, on Oct. 1st ,2004.

AODV through simulations are presented in Section 3.9. Summary is given in Section 3.10.

## 3.2 Cryptographic Primitives for Message Authentication

The design of network layer security is concerned with ensuring that both routing and data messages are exchanged between nodes according to the protocol specifications and its routing states. In this sub-section, we describe two cryptographic primitives that are widely used to authenticate the contents of message exchanged among nodes. *Message authentication* is an essential component in any network layer security design.

### 3.2.1 Hashed Message Authentication Code (HMAC)

A Message Authentication Code (MAC[3]) is a short piece of information used to authenticate a message. The HMAC [5] is a particular type of MAC calculated by using an iterative cryptographic one-way hash function $h$, such as SHA-1 [6] or MD5 [7], in combination with a symmetric secret key. For example, when node $A$ transmits a message to node $B$, it appends to the message an HMAC computed over the transmitted information and shared secret key by using a cryptographic hash function. At reception, node $B$ re-computes the HMAC on the received message using both the same function and key, and checks that the value it obtains equals the HMAC attached to the received message. If the two values match, the message has been correctly received, and the receiver $B$ is assured that the sender is node $A$ with the shared secret key. Since it is computationally infeasible to find an input $x$ given the output $h(x)$ in a cryptographic one-way function $h$, the cryptographic strength of the HMAC depends upon both the underlying one-way function and the size of the secret key. The computation of an HMAC is very efficient and fast, even affordable for low-end devices such as small sensor nodes. However, an HMAC can be verified only by the intended receiver, making it unappealing for broadcast

[3] The acronym "MAC" refers to the Message Authentication Code. To avoid confusion, the term "Medium Access Control" is written out in full in this thesis.

message authentication. Besides, establishing the secret key between any two nodes is not a trivial task [8] in MANETs where the network topology is not known prior to deployment.

### 3.2.2 Digital Signature

A digital signature is used to authenticate the identity of the sender of a message. It ensures that the original content of the message has not been altered. It is typically created through the use of a hash function and asymmetric key cryptography [e.g., Elliptic Curve Cryptography (ECC)] [9]. For example, suppose node *A* wants to send a signed message to node *B*. The first step is to apply a hash function to the message, creating what is called a Message Digest (MD). The job of the hash function is to take a message of arbitrary length and shrink it down to a fixed length. To create a digital signature, node *A* encrypts the MD as opposed to the message itself. Then, node *A* sends to node *B* the encrypted MD and the message. In order to verify the signature, *B* must apply the same hash function as node *A* does to the message, decrypt the encrypted MD using *A*'s public key and compare the two. If they are the same, node *B* can successfully verify the signature.

Since a digital signature involves much more computation overhead in signing/decrypting and verifying/encrypting operations, it is less resilient against Denial of Service (DoS) attacks where an attacker may flood the network with a large number of bogus signatures to exhaust the total computation resources of network for signature verifications. Each node also needs to keep a Certificate Revocation List (CRL) [10] of the revoked certificates. However, a digital signature can be verified by any node when the public key of the source node is known. This makes digital signature scalable to large number of receivers.

## 3.3 Secure Ad hoc Routing Protocols

There are several secure ad hoc routing protocols recently proposed in the literature [4][11]–[15]. In this sub-section, we provide an overview of these protocols and point out the strengths and

weaknesses of each method.

### 3.3.1 Secure AODV (SAODV)

The Secure AODV (SAODV) mechanism proposed in [4] is used to protect the routing messages of the original AODV. SAODV uses digital signatures to authenticate non-mutable fields and hash chains to authenticate the hop-count field in both RREQ and RREP messages.

During the route discovery process, the source node sets a Maximum Hop-count (MHC) to the Time To Live (TTL) value in the Internet Protocol (IP) header, and generates a one-way hash chain of length equal to the MHC plus one with Random Seed Number (RSN) by using the hash function: RSN, $h$(RSN), $h^2$(RSN), … , $h^{\text{MHC-1}}$(RSN), and $h^{\text{MHC}}$(RSN). The source node signs the non-mutable fields of RREQ, $h^{\text{MHC}}$(RSN), and MHC. In addition, the source node includes an element of the hash chain based on the actual hop-count in the RREQ header, that is, $h$(RSN). Since both $h^{\text{MHC}}$(RSN) and MHC are included in the RREQ and authenticated by the signature, an intermediate node can verify that $h^{\text{MHC}-1}[h(\text{RSN})]$ is equal to $h^{\text{MHC}}$(RSN) by applying the hash function MHC – 1 times to $h$(RSN). Before forwarding an RREQ, each node first authenticates the RREQ to ensure that each field is valid. It then increments the hop-count field by one in the RREQ header, hashes the $h$(RSN) and attaches $h^2$(RSN) in the RREQ. Except for the hop-count field and $h^{\text{hop-count}}$(RSN), all other fields of the RREQ are non-mutable and therefore can be authenticated by verifying the signature in the RREQ. When destination node receives an RREQ, it generates an RREP in the same way. SAODV can also allow an intermediate node to generate an RREP by using double signature extension.

During the route maintenance process, SAODV uses the digital signature to protect the RERR message. Both originating and forwarding nodes of the RERR sign the whole message, and thus its neighboring nodes can verify the signature of its previous forwarding node. However, since SAODV does not have a mechanism for authenticating intermediate nodes, malicious attackers can easily join a path and launch various malicious attacks.

### 3.3.2 Ariadne

The Ariadne [11] is a secure routing extension for DSR [16] protocol relying on efficient symmetric cryptography. Assuming the shared secret keys between communicating nodes, the source includes an HMAC computed over non-mutable fields in an RREQ. To ensure that each intermediate node cannot remove the existing nodes from or add extra nodes to the node list in the RREQ, each intermediate node authenticates new information in the RREQ by appending an HMAC of the entire RREQ. Each node uses its own Timed Efficient Stream Loss-tolerant Authentication (TESLA) [17] key chain to compute this HMAC. When the destination determines that the RREQ is valid, it returns an RREP by appending an HMAC computed over non-mutable fields in the RREP. Each intermediate node along the source route appends its TESLA key in the RREP. When the source receives the RREP, it verifies that the end-to-end HMAC and hop-by-hop HMACs are valid. If all these tests give positive results, the source will accept the RREP. However, Ariadne requires global clock synchronization [e.g., using the service of Global Positioning System (GPS)] [18] and each intermediate node needs to wait until it is able to disclose its TESLA key in the route discovery process.

### 3.3.3 Authenticated Routing for Ad hoc Networks (ARAN)

The Authenticated Routing for Ad hoc Networks (ARAN) proposed in [12] provides a solution for secure routing in a managed-open environment where a preliminary certification process is assumed. Route discovery in ARAN is accomplished by a broadcast route discovery message from the source node. The destination node sends a reply to the source node by unicast. The routing message is authenticated at each hop by the previous node's certificate and signature from source to destination. By using unalterable physical metric such as time delay, ARAN avoids attacks against the hop-count field in routing messages. The main limitation of ARAN is that each node must verify multiple signatures for both RREQ and RREP messages. The use of

multiple digital signatures on network-wide broadcast messages can be expensive.

### 3.3.4 Secure Routing Protocol (SRP)

The Secure Routing Protocol (SRP) proposed in [13] attempts to guarantee that the node initiating the route discovery can detect the replies which provide false topological information and can discard these malevolent replies. HMAC is calculated using the shared key between the source and destination and two identifiers. Not only can it validate the integrity of RREQ messages but also authenticate the origin of the packet to the destination. However, this is realized through the existence of a security association between source and destination without the intermediate nodes having to cryptographically validate the control traffic. To limit flooding, each node records the rate at which each neighbor forwards an RREQ and gives a high priority to an RREQ sent through neighbors that less frequently forward RREQs. Like SAODV [4], since SRP does not provide hop-by-hop authentication, a malicious user can join a path and modify the contents of routing messages.

### 3.3.5 Secure Efficient Ad hoc Distance Vector (SEAD)

In the Secure Efficient Ad hoc Distance vector routing protocol (SEAD) proposed in [14], the receiver of the routing update authenticates the sender. Computationally efficient one-way hash functions are used to secure the routing update messages. As it is impossible to invert a one-way hash function, intermediate nodes can only increase the metric in the routing update but cannot decrease it. Therefore, SEAD can secure the lower bound on the hop-count metric in each update message. However, SEAD needs either a shared secret key among each pair of nodes or a broadcast authentication mechanism with synchronized clock to authenticate the source of each routing update messages. As mentioned in Section 3.2.1, establishing the secret key between any two nodes is not a trivial task in MANETs.

### 3.3.6 Secure Link-State Protocol (SLSP)

SLSP (Secure Link-State Protocol) [15] is a secure proactive link-state routing protocol. Link state information is managed by using both Neighbor Lookup Protocol (NLP) and Link State Update (LSU) messages. NLP's HELLO and LSU messages are signed by sender's private key. Thus, all receivers can verify those messages by using sender's public key. A hash chain is used to authenticate the hop-count field of LSU messages similar to SAODV.

## 3.4 Security Threats and Mitigation Requirements for AODV

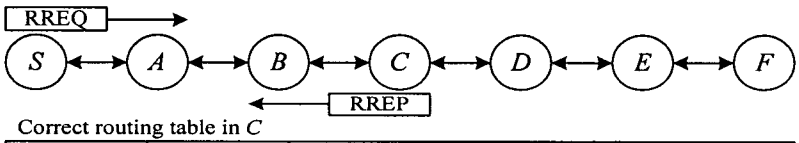
In this section, we analyze the security threats and describe the requirements for AODV routing protocol to mitigate these threats. We call a mobile node or user to be *malicious* if it is an attacker that cannot authenticate itself as a legitimate node due to the lack of valid cryptographic information. On the other hand, we call a mobile node or user to be *compromised* if it is an inside attacker who is behaving maliciously but can be authenticated by the network as a legitimate node and is being trusted by other nodes. We call a mobile node to be *selfish* when it tends to deny providing services for the benefit of other nodes in order to save its own resources. Note that a selfish node is also trusted by other network entities.

### 3.4.1 Attack Models

Several attacks can be launched against the AODV routing protocol as discussed below.

A1. *Message tampering attack*: An attacker can alter the content of routing messages and forward them with falsified information. For example, by reducing the hop-count field in either an RREQ or RREP packet, an attacker can increase its chances to be an intermediate node of the route. On the other hand, a selfish node can relieve the burden of forwarding messages for others by setting the hop-count field of the RREQ to infinity. As shown in Figure 3.1, attacker $C$ can also alter the next hop address $D$ of the corresponding destination $F$ to an unreachable or non-existing address $U$. As a result, data packets passing through node $C$ will never reach their intended destination $F$. In node $C$, routing loops can

also be created by replacing the next hop address *B* to destination *S* either by node *D* or by another rogue node *R*. Attackers can also modify either source or destination addresses in messages to disrupt the operation of AODV routing protocol.




Correct routing table in *C*

| Destination | Next hop | Sequence # | Hop count | Lifetime |
|---|---|---|---|---|
| *F* | *D* | 1 | 3 | current time + Active_Route_Timeout |
| *S* | *B* | 1 | 3 | |

Modified routing table in *C*

| Destination | Next hop | Sequence # | Hop count | Lifetime |
|---|---|---|---|---|
| *F* | *U* | 2 | 2 | ∞ |
| *S* | *R* | 2 | 2 | ∞ |

Figure 3.1 Correct and modified routing tables in AODV.

A2. *Message dropping attack*: Both attackers and selfish nodes can intentionally drop some (or all) routing and data messages. Since all the mobile nodes within a MANET function as both end hosts and routers, this attack can paralyze the network completely as the number of message dropping increases. There are three different control messages in AODV: RREQ, RREP, and RERR. An RREQ dropping attack has no benefit for active attackers as they will not be able to join (or attack) the communication session as an intermediate node. However, a selfish node can save its battery power by dropping an RREQ. An RREP dropping may have a limited impact. Since an intermediate node can also generate an RREP when it has a route to the destination, a source node may receive multiple RREPs. The source will then select one of the available routes. An RREP dropping attack may not always prevent the source to discover a route to the destination; however, it can possibly result in multiple route discoveries when none of intermediate nodes can send an RREP to the destination. Similarly, by dropping an RERR, a compromised user can prevent the

source node from detecting the link breakage. Thus, the source node will continue to send data messages along the broken path resulting in the lost of all messages.

A3. *Falsified message injection attack*: Attackers can impersonate other nodes by altering either their physical or IP addresses and generate falsified messages to disrupt the operation of AODV. An attacker can also falsify messages by changing the information stored in its routing table. For example, as shown in Figure 3.1, when the attacker in node *C* receives an RREQ from its neighbor *B*, it can redirect traffic to itself by unicasting to node *B* an RREP that contains a *destination sequence number* which is greater than the authentic value. Since source node *S* cannot distinguish messages injected by attackers from messages generated by honest nodes, node *C* may then become one of the intermediate nodes of the path. The devastating impacts of falsified message injection attacks on the performance of AODV have been investigated in [19][20] based on simulations. For instance, a single attacker can drop up to 75 % of packets by manipulating destination sequence numbers in some scenarios [19].

A4. *Message replay attack*: Attackers can re-play (or re-transmit) eavesdropped messages again later in a different place. This attack is mainly to consume valuable network resources such as bandwidth or to consume node resources such as memory or computation power. Since neither honest nodes nor central authority can distinguish replayed messages from correct ones, it is difficult to avoid *message replay* attack completely. One well-known type of replay attacks is the *wormhole* attack [21]. A wormhole attacker can tunnel an RREQ directly to a destination node. Since a wormhole attacker may not increase the hop-count, it prevents any other routes from being discovered. The wormhole attack can be combined with the *message dropping* attack.

### 3.4.2 Security Requirements for AODV

In light of our security analyses, the security requirements that are necessary for the AODV

routing protocol include:

1. *Source authentication*: The receiver should be able to confirm that the identity of the source is indeed who or what it claims to be.
2. *Neighbor authentication*: The receiver should be able to confirm that the identity of the sender (i.e., one hop previous node) is indeed who or what it claims to be.
3. *Message integrity*: The receiver should be able to verify that the content of a message has not been altered either maliciously or accidentally in transit.
4. *Access control*: It is necessary to ensure that mobile nodes seeking to gain access to the network have the appropriate access rights.

By providing *neighbor authentication*, malicious users cannot join a communication path. Therefore, they cannot either drop or tamper valid messages. When a source node provides the security services of both *source authentication* and *message integrity* to each message, compromised users can neither impersonate other nodes nor change the non-mutable part of messages. However, compromised users can still manipulate any information stored in its own routing table. Since the routing module is typically implemented as software in the kernel of the operating system, it is possible for a compromised user to modify the routing table if the codes or the associated data are left unprotected. Furthermore, since this attack is launched internally, it cannot be detected either by other layers within the node or by the neighboring mobile nodes. Although mobile nodes are vulnerable to capture or compromise, none of secure on-demand routing protocols described in Section 3.3 can protect the routing table entries from compromised users. To ensure the overall security of the network, we introduce a strong *access control* mechanism that can prevent the routing module from being compromised. In the context of MANET routing, *confidentiality* is not a critical component in non-military applications. Some researchers argue that *non-repudiation* can be used for isolating misbehaving nodes in MANET routing protocols [22]. Table 3.1 shows a comparison between our proposed TRM, STEP, ESTEP,

and SeAODV schemes and the existing secure routing protocols described in Section 3.3. For each scheme, the table indicates the types of attacks that the scheme can protect and those attacks that the scheme is vulnerable to.

Table 3.1 Comparison between different secure on-demand routing schemes

| | Attacks that can be prevented | | Vulnerable attacks | |
|---|---|---|---|---|
| | Compromised | Malicious | Compromised | Malicious |
| SAODV [4] | | A3 | A1,A2,A3,A4 | A1,A2,A4 |
| ARAN [12] | A4 | A1,A2,A3,A4 | A1,A2,A3 | |
| SRP [13] | | A3 | A1,A2,A3,A4 | A1,A2,A4 |
| Ariadne [11] | A1,A4 | A1,A2,A3,A4 | A2,A3 | |
| SeAODV | A4 | A1,A2,A3,A4 | A1,A2,A3 | |
| SeAODV with STEP (or ESTEP) | A1,A3,A4 | A1,A2,A3,A4 | A2 | |
| SeAODV with TRM | A1,A2,A3,A4 | A1,A2,A3,A4 | | |

## 3.5 Network Environments

The secure protocols proposed in this chapter aim to prevent attacks in the network layer. Attacks in other layers (e.g., physical, transport, application) are beyond the scope of this work. Our proposed schemes work under several assumptions. These assumptions are stated as follows:

1. The network links are bi-directional. That is, if node $A$ is able to transmit to node $B$, then node $B$ is also able to transmit to node $A$.
2. There exists a public key infrastructure in the MANET. Each mobile node stores the trusted Certification Authority (CA)'s public key.
3. The ECC [9] is used to generate the digital signature.

We now explain the rationales behind these assumptions. The first assumption is common in practice. Many wireless medium access control protocols require bidirectional links

to exchange several link-layer frames between a source and destination to avoid collisions.

In the second assumption, the CA's public key is used to check the association between the address of a node to the public key of that node. This assumption has also been used in some previous work [4][12] described in Section 3.3. The certificate-based public key distribution is an appropriate concept for MANETs because it does not require online trusted servers. However, it has a problem related to key revocation. That is, each node cannot know whether certificates presented by other nodes have been revoked or not. Solutions to this problem have been proposed by using distributed key management schemes [22][23]. In [22], the service has a public/private key pair that is used to verify/sign public-key certificates of the network nodes. It is assumed that all nodes know this public key. On the other hand, the private key is divided into $n$ shares using a *threshold cryptography* scheme, and the shares are assigned to $n$ arbitrarily chosen nodes, called servers. For the service to sign a certificate, each server generates a partial signature for the certificate using its private key share and submits it to a combiner that computes the signature from the partial signatures. In [23], each node signs certificates for other nodes without an online CA. When a node $x$ wants to obtain the authentic public key of a node $y$, it acquires a chain of certificates such that the first certificate of the chain can directly be verified by its own public key, each remaining certificate can be verified using the public key in the previous certificate of the chain, and the last certificate contains the public key of the target node $y$. This scheme allows each node to obtain the public keys from other nodes in the network.

The third assumption aims to reduce the overhead in terms of delay and bandwidth. The use of ECC reduces the time to generate and verify the signature when compared with the RSA (Rivest, Shamir, Adleman)'s algorithm. Moreover, ECC with a 160-bit key offers the same security as the RSA system with a 1024-bit key. As a result, the length of the public key and private key is shorter by using ECC than RSA. For example, under a specific set of conditions [24], the delay for signature generation and verification for a 1024-bit RSA key is about 60 ms

whereas the delay is only 6 ms for an ECC 160-bit key.

## 3.6 Tamper Resistant Module (TRM)

Although our proposed SeAODV in Section 3.8 can defend AODV against malicious users who do not possess valid cryptographic information and cannot authenticate themselves as legitimate nodes, it is still vulnerable to some attacks by compromised users who are authenticated by the network and trusted by other nodes. Since a compromised user can join a valid path and decide the next hop address based on its own local routing table in AODV, the compromised user can still either drop all messages or alter the next hop address of the corresponding destination to an unreachable or non-existing address as described in Section 3.4.1. To ensure the overall security of the network, we introduce a strong *access control* mechanism through the deployment of Tamper Resistant Modules (TRMs) [1] that can prevent the routing module from being compromised. We describe the functionality, implementation, and limitations of the proposed TRMs.

### 3.6.1 Functionality

We propose the use of TRM to protect the routing module. A TRM protects system software/hardware in a mobile node from being modified, and prevents secret embedded information from being extracted by an attacker. Most attempts to modify a TRM would cause either no change in its behavior or a loss of functionality [25]. Therefore, TRM can be employed to protect routing tables and secret information stored in a mobile host against tampering by the end user. There are two types of TRM: Tamper Resistant Hardware (TRH) and Tamper Resistant Software (TRS).

TRH is currently widely used in smartcards [26]. Sensitive information is kept in the Electrically Erasable Programmable Read-Only Memory (EEPROM) together with several kilobytes of executable code. Smartcard can be considered as a safe containing a microcomputer

that performs all the relevant cryptographic operations. This safe has lid switches and circuitry which interrupts power to memory, thus erasing key material, when the lid is opened.

On the other hand, TRS aims to protect software algorithms. There are several techniques [25][27][28] of generating TRS code, which prevents an attacker from illegal use of software packages or modification of the codes. Such secure software is resistant to reverse engineering and keeps an encryption/decryption algorithm or data secret unless specialized hardware analysis tools are used [27]. The basic technique of generating TRS code is to convert the instruction patterns useful for program analysis into un-analyzable instruction patterns without changing the original algorithm.

### 3.6.2 Implementation

Figure 3.2 shows the Linux implementation of the Internet protocol address family as a series of connected layers of software. In this reference model, the routing module is implemented in the kernel of the operating system while the medium access control layer functionalities are implemented in a Network Interface Card (NIC).

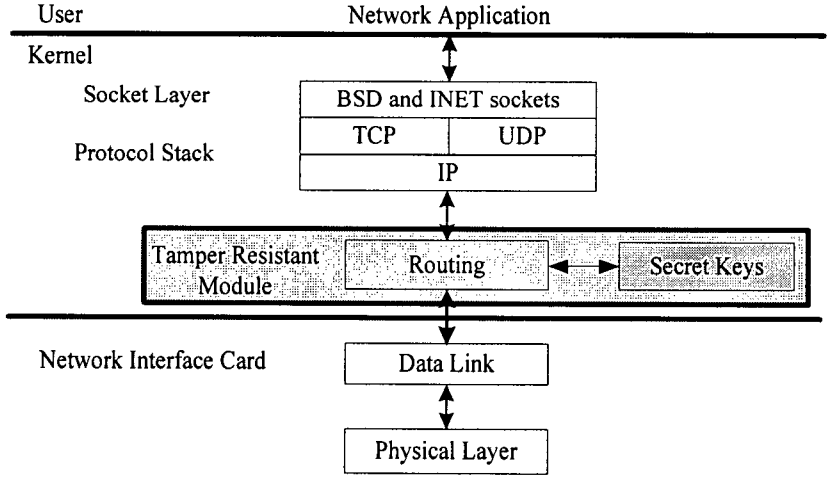



Figure 3.2 Tamper resistant module in Linux.

There have been several software implementation studies of ad hoc routing protocol in real world environments [29][30]. As a practical example, we can use the model of AODV-UCSB [29] as shown in Figure 3.3. In this reference model, to trigger SeAODV protocol events such as route discovery, a SeAODV daemon is installed in user space, which is a common design in moderm operating systems. The Netfilter [31] kernel module is installed for extending the kernel functionalities. Netfilter is a set of hooks inside the Linux kernel that allows kernel modules to register callback functions with the network stack. A registered callback function is invoked for every packet that traverses the respective hook within the network stack. Lastly, the SeAODV kernel module, which can examine, drop, discard, modify or queue the packets for the SeAODV daemon, is installed in the kernel space of the operating system. The main advantage of this implementation is that there is no need to recompile the complete kernel source code again. In other words, the Netfilter, SeAODV kernel module, and SeAODV daemon can be installed independently without recompiling or rebooting a running kernel.

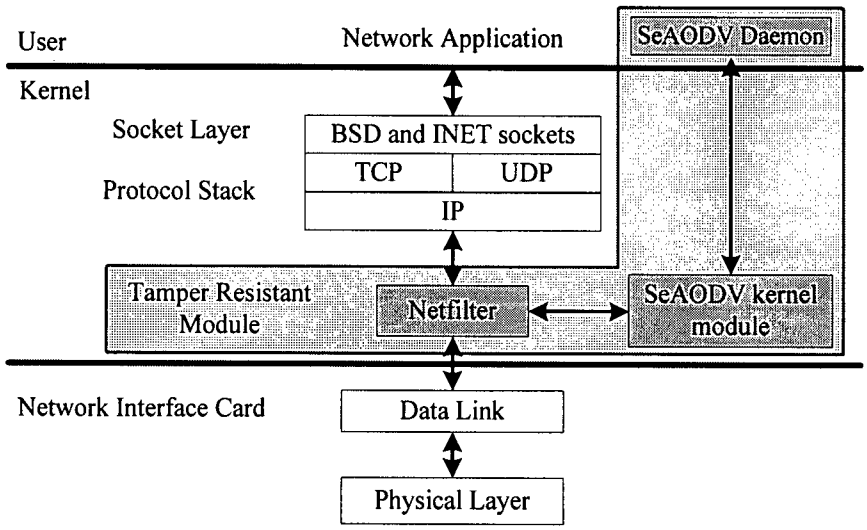



Figure 3.3 TRS implementation in Linux.

For the distribution of TRS, a trusted CA is required whose public key is known to all nodes. In addition, this CA is required to have a TRS code generator [25], which converts the

original source code into TRS code. Before joining the MANET, each mobile node must request a certificate and TRS code from the CA (either offline or online). Each node receives exactly one certificate and TRS code after securely authenticating its identity to the CA. The methods for secure download [32] of software can be used to verify the integrity of the downloaded software in this process.

For example (see Figure 3.4), a mobile node sends its unique node ID and the type of routing protocol to the CA. In return, the CA sends the mobile node (i) the TRS code of the corresponding protocol with the embedded secret information if necessary and (ii) a certificate that consists of the public key of the node with the digital signature of the CA. The corresponding mobile node then compiles and runs the TRS code. Thus, each node's routing protocol is now tamper resistant. There are several techniques for the construction of TRS code proposed in the literature [25][27][28], a detailed discussion of these techniques is beyond the scope of this thesis.

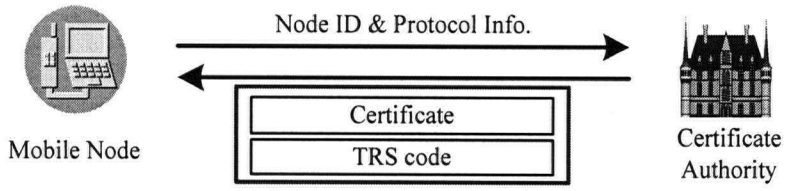



Figure 3.4 TRS exchange between mobile node and CA.

### 3.6.3 Limitations

A tamper resistant system protects against unauthorized attempts to read or modify the contents. The use of TRM can be justified due to the seriousness of table tampering attacks in hostile environments such as military battlefield and rescue site. However, although the tamper resistant technology has the appealing features as described above, it is still possible to attack this system by using special equipment. Interested readers can refer to [33] for details. Therefore, the perfect

proof against tampering is improbable. Moreover, TRM cannot protect the routing module from software bugs and hardware failures. In the next section, we investigate an alternative security mechanism that can defend against routing table tampering attack without assuming the deployment of TRM.

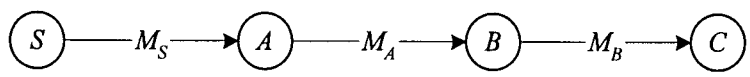
## 3.7 Secure Routing Table in AODV

In this section, we describe a security mechanism that protects the routing table entry in AODV routing protocol. We use the following notations in this chapter:

1. $Sign_A(M)$ denotes that a message $M$ is digitally signed by the private key of node $A$.
2. $< M_1 \| M_2 >$ denotes that message $M_1$ is concatenated with message $M_2$.

### 3.7.1 Secure Table Entry Protection (STEP)

Recall that AODV is vulnerable to routing table tampering attacks because node $B$ cannot verify the correctness of the *hop-count field* of routing control messages received from its neighboring node $A$ (see Figure 3.5). Moreover, since AODV allows intermediate nodes to generate an RREP with their known sequence number for the destination, attackers can manipulate the *destination sequence number*. To address this problem, $A$ must prove the correctness of both the *originator sequence number* of node $S$ and *hop-count field* in a control message to its downstream neighbor $B$.




$M_S$ = <MESSAGE, $ts_S$, $Sign_S(sn_S \| hc_{SS} \| ts_S)$>

$M_A$ = <MESSAGE, $ts_S$, $Sign_S(sn_S \| hc_{SS} \| ts_S)$, $ts_A$, $Sign_A(hc_{AS} \| ts_A)$>

$M_B$ = <MESSAGE, $ts_S$, $Sign_S(sn_S \| hc_{SS} \| ts_S)$, $ts_A$, $Sign_A(hc_{AS} \| ts_A)$, $ts_B$, $Sign_B(hc_{BS} \| ts_B)$>

Figure 3.5 Example for STEP with digital signature.

Our proposed STEP mechanism provides authentication for both the *originator*

*sequence number* and *hop-count field* in the routing control messages. The receiving node can confirm the correctness of control messages by verifying the digital signatures of the two consecutive upstream nodes and the source node (i.e., originator), which created messages. As shown in Figure 3.5, whenever node $S$ sends a message to its neighbor, it needs to compute the digital signature for the *originator sequence number* ($sn_S$), *hop-count* ($hc_{SS}$) and *timestamp* ($ts_S$) fields. This digital signature is then appended at the end of the message. Therefore, node $S$ sends a message $M_S$ = <MESSAGE, $ts_S$, $Sign_S(sn_S||hc_{SS}||ts_S)$> where MESSAGE denotes either the original AODV's RREQ or RREP, and $hc_{SS} = 0$.

When one of its neighbors, node $A$, receives this message, it verifies the digital signature, signs the updated hop-count field with a timestamp, and then forwards the modified message $M_A$ to its neighbor $B$ (see Figure 3.5). The value of $hc_{AS}$ (or $hc_{BS}$) denotes the hop distance from node $A$ (or $B$) to $S$ [i.e., 1 (or 2)]. In this way, every node except the destination node attaches two additional fields [i.e., *digital signature* (40 bytes) [9] and *timestamp* (4 bytes)] in each packet, and forwards it to its neighbor. Thus, any malicious change in the sequence number on $S$ will be detected by the attacker's one-hop downstream node by verifying these signatures.

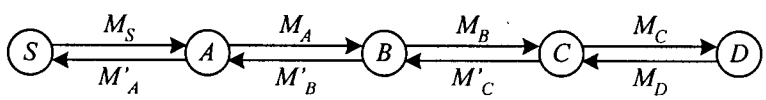
Moreover, the receiving node can verify the correctness of the hop-count field. For example, if the previous node has not increased the value in the hop-count field by one, the signature of the two-hop upstream neighbor cannot be verified correctly by substituting (*hop_count* – 1) for the value of hop-count field. *If there are no colluding nodes in the network, the two digital signatures for two upstream nodes are sufficient to verify the correctness of hop-count field in the received message.* Therefore, each intermediate node's signature is forwarded only up to two-hop downstream nodes. We will discuss the colluding attackers in Section 3.7.4

### 3.7.2 Route Discovery with STEP

We now present the AODV's route discovery scheme based on STEP. The intuition behind this

secure route discovery is to make both destination sequence number and hop-count field to be verifiable. Thus, routing table tampering attacks can be detected in the network.

Before node $S$ sends an RREQ message to node $D$, it needs to compute the digital signature for the sequence number, hop-count field, and timestamp, and then append both *digital signature* and *timestamp* to the message as shown in Figure 3.6. When an intermediate node $A$ receives this message, it needs to verify the digital signature. If the digital signature is valid, it will update the reverse path to node $S$ and broadcast this packet with its signature and timestamp to its neighbors again. Eventually, an RREQ message will reach destination $D$. Node $D$ can then verify the digital signatures of $S$, $B$, and $C$ (see Figure 3.6). If an RREQ message is valid, each node will store both the routing information with IDs and all digital signatures in its routing table.




$M_S$ = <RREQ_, $ts_S$, $Sign_S(sn_S \| hc_{SS} \| ts_S)$>

$M_A$ = <RREQ_, $ts_S$, $Sign_S(sn_S \| hc_{SS} \| ts_S)$, $ts_A$, $Sign_A(hc_{AS} \| ts_A)$>

$M_B$ = <RREQ_, $ts_S$, $Sign_S(sn_S \| hc_{SS} \| ts_S)$, $ts_A$, $Sign_A(hc_{AS} \| ts_A)$, $ts_B$, $Sign_B(hc_{BS} \| ts_B)$>

$M_C$ = <RREQ_, $ts_S$, $Sign_S(sn_S \| hc_{SS} \| ts_S)$, $ts_B$, $Sign_B(hc_{BS} \| ts_B)$, $ts_C$, $Sign_C(hc_{CS} \| ts_C)$>

$M_D$ = <RREP_, $ts_D$, $Sign_D(sn_D \| hc_{DD} \| ts_D)$>

$M'_C$ = <RREP_, $ts_D$, $Sign_D(sn_D \| hc_{DD} \| ts_D)$, $ts'_C$, $Sign_C(hc_{CD} \| ts'_C)$>

$M'_B$ = <RREP_, $ts_D$, $Sign_D(sn_D \| hc_{DD} \| ts_D)$, $ts'_C$, $Sign_C(hc_{CD} \| ts'_C)$, $ts'_B$, $Sign_B(hc_{BD} \| ts'_B)$>

$M'_A$ = <RREP_, $ts_D$, $Sign_D(sn_D \| hc_{DD} \| ts_D)$, $ts'_B$, $Sign_B(hc_{BD} \| ts'_B)$, $ts'_A$, $Sign_A(hc_{AD} \| ts'_A)$>

Figure 3.6 Example for route discovery (RREP from node $D$) where RREQ_ and RREP_ denote the original AODV's RREQ and RREP, respectively. Note that original AODV's messages have the fields of the destination sequence number and hop-count.

When the RREQ message arrives at the destination or to an intermediate node, which has an entry in its cache, an RREP message is being sent. Depending on whether the node is the destination $D$ or an intermediate node, which has a fresh route to node $D$, the processing is

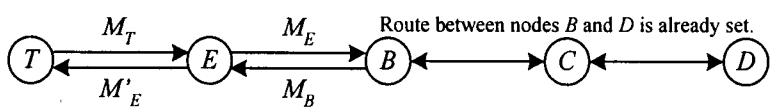
slightly different. These two scenarios are described in the following subsections.

#### 3.7.2.1 RREP from the destination

Figure 3.6 shows an example of secure route discovery. In this example, only destination $D$ sends an RREP message to the corresponding RREQ message. Once created, the RREP message is sent to the next hop toward the originator of the RREQ message according to its local routing table entry for that originator. When the source $S$ or intermediate nodes $A$, $B$, and $C$ receive an RREP message, they will verify all digital signatures in the RREP message. If the message is valid, they will store both the routing information with IDs and those signatures, and update their forward route to the destination $D$ using the neighbor from which they receive the RREP message.

#### 3.7.2.2 RREP from the intermediate node

Consider the example in Figure 3.7 where an RREP is generated by an intermediate node $B$. In the RREP, node $B$ attaches three digital signatures of $D$, $C$, and itself with timestamps. If this message is valid, node $E$ can confirm that the received RREP from $B$ is correct, up-to-date, and has not been modified by the table tampering attackers. Thus, it will update and forward the RREP to node $T$.




$M_T$ = <RREQ_, $ts_T$, $Sign_T(sn_T \| hc_{TT} \| ts_T)$>

$M_E$ = <RREQ_, $ts_T$, $Sign_T(sn_T \| hc_{TT} \| ts_T)$, $ts_E$, $Sign_E(hc_{ET} \| ts_E)$>

$M_B$ = <RREP_, $ts_D$, $Sign_D(sn_D \| hc_{DD} \| ts_D)$, $ts'_C$, $Sign_C(hc_{CD} \| ts'_C)$, $ts'_B$, $Sign_B(hc_{BD} \| ts'_B)$>

$M'_E$ = <RREP_, $ts_D$, $Sign_D(sn_D \| hc_{DD} \| ts_D)$, $ts'_B$, $Sign_B(hc_{BD} \| ts'_B)$, $ts'_E$, $Sign_E(hc_{ED} \| ts'_E)$>

Figure 3.7 Example for route discovery (RREP message from node $B$).

### 3.7.3 Route Discovery with Efficient STEP (ESTEP)

The main limitation of STEP is that each node must verify multiple signatures (up to three) for both RREQ and RREP packets. The use of multiple digital signatures on a network wide broadcast RREQ packets can be very expensive. To this end, we propose an *Efficient STEP* (*ESTEP*), which can avoid the use of expensive multiple digital signatures on an RREQ by using a single signature.

When node $S$ sends an RREQ message to node $D$, it needs to compute the digital signature for *originator sequence number* and *timestamp*, and then appends this message at the end: $M_S$ = <RREQ_, $ts_S$, $Sign_S(sn_S||ts_S)$>. When one of its neighbors, node $A$, receives this message, it can verify the digital signature to check both the authenticity and the integrity. If the digital signature is valid, it will update the temporal reverse path to node $S$ and broadcast this packet $M_A$ to its neighbor $B$ again and so on, where the message $M_A = M_S$. Note that the other fields are updated according to AODV routing protocol. Eventually, the RREQ message will reach the destination $D$.

If this message is valid, node $D$ will generate the temporal path to node $S$. Since intermediate and destination nodes (i.e., $A$, $B$, $C$, and $D$ in Figure 3.7) can verify neither *hop-count* nor *next hop address* field in its routing entry corresponding to node $S$, they must not generate an RREP message by themselves on behalf of node $S$ in response to any RREQ message.

After receiving a valid RREQ message, the destination node $D$ creates an RREP message. The RREP message is sent to the next hop toward the originator of the RREQ message. Unlike an RREQ message, the RREP message needs to be relayed according to the STEP mechanism described in Section 3.7.1 with multiple signatures. There is one additional field in RREP message: the hop-count field from node $S$ to $D$. This field is signed by destination node $D$, and verified by source node $S$. If this value is matched with that of hop-count field in the RREP message, this unidirectional path from source $S$ to destination $D$ can be trusted as the path without routing table tampering attackers. On the other hand, destination $D$ can trust the routing

path to source $S$ after it receives the first data packet from the source $S$.

### 3.7.4 Extension against Colluding Attackers

STEP is effective against routing table tampering attacks from non-colluding users (i.e., individual compromised user). However, it cannot prevent colluding attackers (i.e., a set of compromised users) from cheating the hop-count field. To prevent $n$ serial colluding nodes from decreasing or not increasing the hop-count to node $S$ in a message, each node must have $(n+1)$ signatures on the hop-count field starting from its one-hop neighbor to $(n+1)$ hops upstream node toward node $S$. For example, suppose nodes $A$ and $B$ are colluding routing table tampering attackers (i.e., $n = 2$). When node $A$ receives a message from node $S$, it will not increase the hop-count field and forward to its neighbor $B$. Node $B$ may increase the hop-count field and relay again to its neighboring node $C$. Since node $C$ verifies three (i.e., $n+1 = 3$) signatures from its three upstream nodes $S$, $A$, and $B$, it can prove that both nodes $A$ and $B$ are tampering the message from $S$. This signature can also be used for non-repudiation purpose.

### 3.7.5 Integrating STEP with Secure Routing Protocol

Due to the use of multiple signatures, STEP can introduce significant routing overhead relative to the AODV routing protocol. It may create the scalability problem and degrade the network performance. However, since STEP is designed only to protect routing table entries from compromised users, it can be invoked only when necessary. For example, when STEP is used together with a secure on-demand routing protocol such as SAODV [4] or our proposed SeAODV (see Section 3.8), a node can find a route using those routing protocols. If the source node cannot find a valid route due to malicious changes on either destination sequence number or hop-count, it will set a flag indicating that it activates the use of STEP (or ESTEP). In this way, STEP can still be considered as efficient in the large and dense networks.

## 3.8 Proposed Secure AODV (SeAODV) Protocol

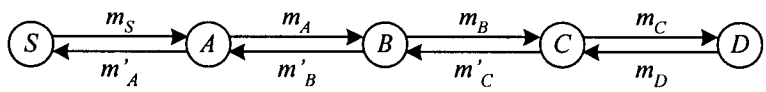
In this section, we present the proposed SeAODV protocol, a secure routing extension to the original AODV routing protocol. SeAODV aims to secure routing control messages from attackers described in Section 3.4.1. We then propose a Secure Data Forwarding (SDF) scheme based on SeAODV for secure transmissions of data packets over the wireless links. SDF aims to maintain the integrity of data messages and prevents data dropping attacks.

### 3.8.1 Secure Route Discovery

Similar to [4][12], we use *digital signature* to provide source authentication and to protect the integrity of the non-mutable part of an RREQ, but with the following differences. We do not protect the integrity of the mutable part (i.e., a hop-count field) of an RREQ. An example is shown in Figure 3.8. When a node $S$ sends an RREQ to node $D$, it needs to compute the digital signature for non-mutable fields (i.e., $n_S$ in Figure 3.8) with its own private key, and then appends this at the end of message. When one of its neighbors, node $A$, receives a message $m_S$, it can verify the digital signature to check the integrity of the non-mutable part of an RREQ. If the digital signature is valid, it updates the reverse path to node $S$ and broadcasts updated message $m_A$ to its neighbor $B$ again and so on. Eventually, an RREQ will reach the destination $D$.

After receiving a valid RREQ, the destination node $D$ creates an RREP. The RREP is unicast to the next-hop toward the originator of RREQ according to its local routing table entry for that originator $S$. Unlike RREQ, the RREP needs to be relayed with the *double signatures*, i.e., two digital signatures are applied. The first one is for the non-mutable part (e.g., source and destination addresses), and the second one is for the mutable part (i.e., hop-count field). As the RREP is sent towards $A$, the hop-count field is incremented by one at each hop, and the digital signature of the mutable part is updated and signed by intermediate node $C$ with its own private key as shown in Figure 3.8. Note that there is one additional field: the hop-count field in the corresponding RREQ from node $S$ to $D$ (i.e., $hc_{DS}$ in Figure 3.8). This field is signed by destination node $D$ together with non-mutable fields, and verified by source node $S$. If this value

is matched with that of hop-count field in the received RREP, this unidirectional path from source $S$ to destination $D$ can be trusted as the path without malicious attackers (except replay attackers). On the other hand, destination $D$ can trust the routing path to source $S$ after it receives the first data packet from the source $S$.




$m_S$ = < RREQ_, $Sign_S(n_S)$, $hc_{SS}$ >
$m_A$ = < RREQ_, $Sign_S(n_S)$, $hc_{AS}$ >
$m_B$ = < RREQ_, $Sign_S(n_S)$, $hc_{BS}$ >
$m_C$ = < RREQ_, $Sign_S(n_S)$, $hc_{CS}$ >
$m_D$ = < RREP_, $hc_{DS}$, $Sign_D(n_D||hc_{DS})$, $Sign_D(hc_{DD})$ >
$m'_C$ = < RREP_, $hc_{DS}$, $Sign_D(n_D||hc_{DS})$, $Sign_C(hc_{CD})$ >
$m'_B$ = < RREP_, $hc_{DS}$, $Sign_D(n_D||hc_{DS})$, $Sign_B(hc_{BD})$ >
$m'_A$ = < RREP_, $hc_{DS}$, $Sign_D(n_D||hc_{DS})$, $Sign_A(hc_{AD})$ >

| | | |
|---|---|---|
| - non-mutable fields of RREQ_ at node $S$ | = | $n_S$ |
| - non-mutable fields of RREP_ at node $D$ | = | $n_D$ |
| - signature of node $X$ on message $m$ | = | $Sign_X(m)$ |
| - hop-distance between node $X$ and $Y$ | = | $hc_{XY}$ |

Figure 3.8 Secure route discovery with digital signatures where RREQ_ and RREP_ denote the original AODV's RREQ and RREP, respectively.

Note that when an intermediate node receives an RREQ and has a routing cache for that destination, the AODV standard allows an intermediate node to send an RREP to the source node directly. However, since source node cannot trust the routing information generated by an intermediate node, the intermediate node does not send RREP but simply forwards the RREQ to its neighbors in our scheme. Thus, an attacker cannot inject a falsified RREP by changing the information stored in its own routing table (i.e., hop-count, destination sequence number) as described in Section 3.4.1.

Since a falsified RREQ from compromised users can flood the whole network, injecting many RREQs results in a DoS attack. To reduce the power of this attack, the rate of generating RREQs should be limited. Thus, neighboring nodes can filter out excessive RREQs immediately.

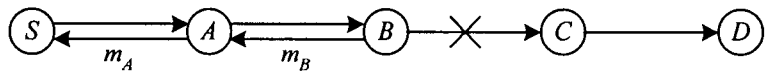
### 3.8.2 Secure Route Maintenance

In AODV, a node can initiate the generation of an RERR in several situations [3]. Consider the case when a node detects a link breakage for the next-hop of an active route in its local routing table while transmitting data. For example, when an upstream node has not received the link layer acknowledgement (ACK) after several retransmissions, it will declare this link to be broken and send an RERR to its neighboring node.

A problem arises when the downstream node is selfish and launches a *message dropping* attack. The upstream node cannot distinguish whether the link breakage is due to mobility of the downlink node or intentional message dropping. We resolve this issue by proposing an extension of the RERR to include the addresses of the two end nodes of the broken link. As shown in Figure 3.9, the generator $B$ of an RERR signs the original RERR and the addresses of the two suspicious nodes separately for authentication and integrity. Node $B$ then sends it to its neighbor $A$ that belongs to a precursor list of unreachable destination $D$. When a node $A$ receives this message, it verifies the signature for the original RERR. If that digital signature is valid, node $A$ will re-generate a signed RERR with its own private key, and attach the addresses of the two suspicious nodes and the corresponding node $B$'s signature.

When the source node receives the RERR, it will re-initiate the route discovery process, and have a cache to store the addresses of these two end nodes of the broken link. Since the reception of multiple consecutive RERRs from the same link is a rare instance, a source node increases the suspicious value of two intermediate nodes by one when it receives two consecutive RERRs from the same broken link within a certain period. This suspicious value will not be reported to any other node in order to avoid blackmail attacks, which refers to the false

report of a good node as a bad one. The suspicious values of these two nodes are decreased by one after the source node has not received an RERR with the addresses of these two suspicious nodes for a certain period of time.




$m_B$ = < RERR$_B$, $Sign_B$(RERR$_B$), $Addr_B$, $Addr_C$, $Sign_B(Addr_B||Addr_C)$ >

$m_A$ = < RERR$_A$, $Sign_A$(RERR$_A$), $Addr_B$, $Addr_C$, $Sign_B(Addr_B||Addr_C)$ >

- medium access control address of node $X$ = $Addr_X$

Figure 3.9 Example of an RERR generation for suspicious node detection where RERR$_X$ denote the original AODV's RERR generated by node $X$.

The source node will append the addresses of nodes, which have reached the maximum suspicious value $\theta$, in the RREQ for route discovery later. Those suspicious nodes will not be chosen as part of a new route from this source node to any destination. By using the secure route maintenance mechanism of SeAODV, we can protect the RERR from *message tampering* attackers and avoid suspicious nodes from joining a path.

### 3.8.3 Consideration of Control Message Dropping Attacks

A selfish node can drop an RREQ to save its resource from forwarding messages for others. While this service disruption attack is possible in all secure ad hoc routing protocols [4][11]–[15], RREQ dropping attacks cannot subvert the operation of routing protocols. On the other hand, since the intermediate nodes cannot generate an RREP in SeAODV, a source node may receive a single RREP from a destination node. Although a malicious user cannot join the authenticated path for a real communication in SeAODV, it can relay an RREQ with a modified hop-count field to attract an RREP from a destination node. Since it cannot provide *neighbor authentication*

to its upstream node towards the source node, it may drop this RREP to fail the route discovery process. Moreover, a selfish node can also drop an RREP to save its resource. Thus, an RREP dropping attack may possibly result in multiple route discoveries. To reduce the impacts of RREP dropping attack, we can extend SeAODV with the path disjoint multipath routing protocol, such as AOMDV [34]. There are two different types of path disjoint routes: link-disjoint and node-disjoint routes. Path disjointness has the nice property that paths fail independently. In general, node-disjointness is a more strict condition than link-disjointness, and thus may give a lower number of disjoint routes. However, node-disjoint multipath is advantageous when there are attackers in the networks. This is because link-disjoint path cannot avoid either attackers or selfish nodes, which have several independent links with multiple neighbours.

An RERR dropping attack has no benefit for the selfish node. Although an RERR can be dropped due to transmission failures or other kind of failures [35] in wireless channels, the forwarding node will reinitiate processing for an RERR whenever it receives a message destined to unreachable destinations. However, our protocol described so far cannot detect either RERR or data dropping attack by a compromised user that is located in the discovered path. To solve this problem, we can use the reputation system [35] to identify and isolate attackers from the route used. One another possible solution is to introduce a feedback message from destination to the source to inform the performance of the path periodically [11].

### 3.8.4 Consideration of Replay Attacks

Unlike other malicious attackers, a replay attacker can join a valid path discovered by SeAODV. As shown in Figure 3.10, attacker *M* can replay eavesdropped RREQ and RREP between nodes *B* and *C*. After joining a path, the attacker *M* can increase the end-to-end delay or drop messages. To avoid replay attackers from joining a path, we introduce the 32 bits timestamp value in all RREPs with a limited clock skew. When generating an RREP, destination node *D* signs the timestamp with its mutable field and forwards this to its next hop node *C*. When *C* receives this

message, it checks whether the timestamp in the RREP is valid. If the timestamp has been generated within a given threshold value (for example, the maximum transit time between two-hop nodes), it forwards an updated RREP with a new timestamp to its neighbor *B* again and so on. Thus, we can prevent replay attackers from joining a path. Note that we do not use the timestamp in an RREQ because all valid paths from source to destination are secured by relaying an RREP as shown in Section 3.8.1.

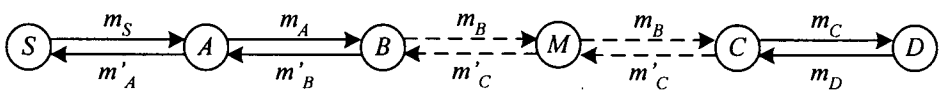



Figure 3.10 A replay attack in route discovery process.

### 3.8.5 Secure Data Forwarding (SDF) based on SeAODV

A malicious attacker can impersonate a downstream node and send a falsified link layer ACK to its upstream node to prevent the initiation of route maintenance process. For example, assume that a path exists between source *S* and destination *D* through intermediate nodes *A*, *B*, and *C* as illustrated in Figure 3.11(a). When a link breakage happens between nodes *B* and *C* due to mobility of *C* as shown in Figure 3.11(b), the upstream node *B* cannot receive a link layer ACK from *C* after sending the data message. After the maximum number of retransmission attempts, node *B* can detect the lost of the link to *C* and send an RERR according to its precursor list. However, a malicious attacker *M* can prevent the generation of an RERR by impersonating *C*. To start this attack, *M* changes its physical address to match *C*'s, moves closer to *B*. It then sends an ACK to *B* as illustrated in Figure 3.11(c). Since *B* cannot detect its link breakage with *C* due to a falsified ACK, it will keep sending data messages. As a result, all data messages may be dropped in this broken link.

In this section, we propose a Secure Data Forwarding (SDF) mechanism to authenticate data messages and to prevent the generation of a falsified link layer ACK. Applying an asymmetric authentication method for all messages such as the digital signature may not be suitable due to the high computational power required to generate and verify the digital signature for each data message. Therefore, we propose the use of shared symmetric key between neighbors on the path.

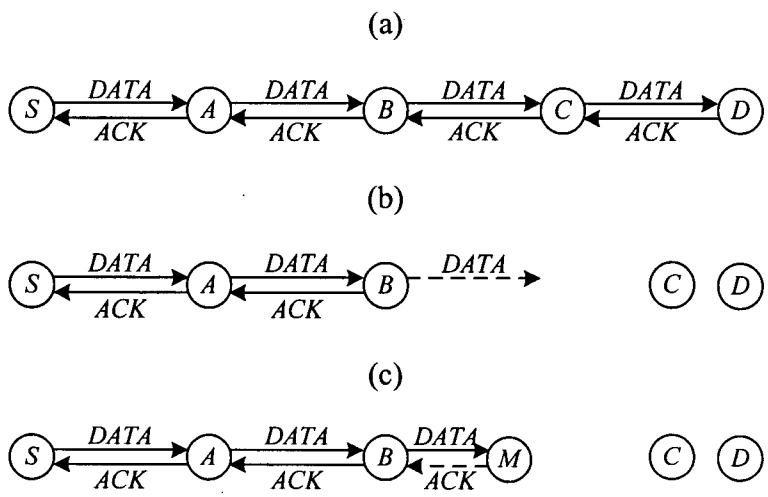



Figure 3.11 A sequence of events that generates a falsified link layer ACK.

Since each participating node of the route from source to destination has to exchange RREQ and RREP during the route discovery phase, we take advantage of this interaction for key exchange, using an authenticated Diffie-Hellman (DH) key exchange protocol [36] to generate the symmetric keys (e.g., $K_{AB}$ in Figure 3.12) between neighbors by exchanging authenticated public values $X_A$ and $X_B$. Both source and destination can also generate the symmetric key for end-to-end authentication. The detail operations for an authenticated DH key exchange method can be found in [36].

Considering the example in Figure 3.12, the upstream node $A$ is the sender while the downstream node $B$ is the receiver. Note that a generated key $K_{AB}$ is different on each link along

the path. The data packet integrity is maintained by using an HMAC [5]. Each SDATA (Secure DATA) frame includes an HMAC value (16 bytes in MD-5 [7]) that is a function of the data, the previous HMAC value, and the symmetric key. Each SACK (Secure ACK) frame also includes an HMAC value that is a function of the ACK, the previous HMAC value, and the symmetric key. The detail operations for HMAC can also be found in Section 3.2.1. With the use of HMAC, each node can verify the integrity of data and ACK messages in the path.

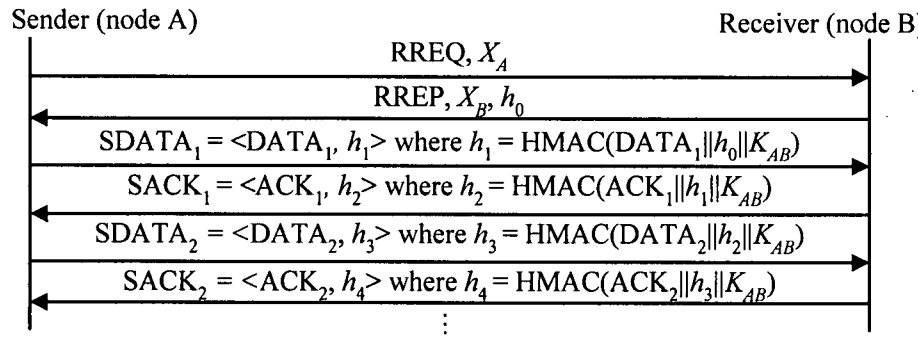



Figure 3.12 Hop-by-hop data integrity check where $h_0$ is the random initial value.

### 3.8.6 Comparison with SAODV

Comparing our proposed SeAODV with the SAODV [4], we note that both protocols use the digital signature for the non-mutable part in both RREQ and RREP messages. On the other hand, they use different schemes to protect the mutable part. SAODV uses hash chains to authenticate the hop-count field of RREQ and RREP. However, since SAODV does not have a mechanism for authenticating intermediate nodes, malicious attackers can easily join a path and launch various malicious attacks. For example, an attacker may drop messages or not increase the hop-count field to make other nodes believe that this is a shorter path to destination. On the other hand, SeAODV can verify each node along the path by signing the mutable field of an RREP at each hop. Therefore, malicious attackers have no opportunity to redirect traffic toward itself.

SeAODV can also prevent message replay attack by including a timestamp with an RREP with a limited clock skew.

## 3.9 Performance Comparisons

We use the network simulator (*ns*2) to compare the performance among STEP, ESTEP, and the original AODV routing protocol. We also compare the performance among SeAODV with SDF, SAODV, and the original AODV routing protocol. In this section, we refer to SeAODV with SDF simply as SeAODV.

The network topology consists of 50 nodes randomly placed over a 1000×1000 $m^2$ flat-grid, and each simulation run takes 900 simulated seconds in Figures 3.19–3.33. We assume that 15 of these nodes are CBR sources, each sending fixed size 512-byte packets at a rate of 4-packets/sec. A random waypoint model is used for the mobility model at a speed uniformly distributed from 0 to 20 m/s. The physical characteristics of each mobile node's radio interface approximate the Lucent WaveLAN, operating as a shared-medium radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 m. The propagation model combines a free space propagation model and a two-ray ground reflection model. Other parameters are the same as in ns-2 version b8a [37].

For both SAODV and SeAODV, we use ECC with a 20 byte key and 40 byte signature [9]. The digital signature generation delay and verification delay are assumed to be 2.0 ms and 4.0 ms, respectively. These delay values are based on the measurements in [24]. Our results are based on simulation over 10 runs, and the error bars represent the 95 % confidence interval of the mean in Figures 3.13–3.33. The packet sizes of RREQ and RREP in the original AODV [3] are 24 and 20 bytes, respectively. Note that the size of RERR is a variable which depends on the number of unreachable destinations. In SeAODV, the size of additional fields of RREQ, RREP, and RERR are 68 bytes [i.e., one signature (40), public value $X_A$ (16) (see Figure 3.12), and two physical addresses (12)], 113 bytes [i.e., 2 signatures (80), $X_B$ (16), $h_0$ (16) (see Figure 3.12), and

hop-count (1)], and 52 bytes [i.e., signature (40), and two physical addresses (12)], respectively. In SAODV, the size of additional fields of RREQ, RREP, and RERR are 76 bytes [i.e., one signature (40), top hash (16), hash (16), and header (4)], 76 bytes (same with RREQ), and 44 bytes [i.e., one signature (40) and header (4)], respectively. In this simulation, the maximum suspicious value $\theta$ is set to one. This means when source node receives consecutive RERRs from the same link, these two end nodes of the broken link will not be chosen as part of a new route from this source node to any destination later.

We use six different metrics for performance evaluations:

1. *Packet delivery fraction*: The measured ratio of the number of data packets delivered to the destinations to the number of packets generated by all traffic sources. This metric indicates the ability of the protocol to discover routes.
2. *Normalized routing overhead* (*bytes*): The ratio of overhead bytes to delivered data bytes. Note that each hop-wise packet transmission is counted as one packet transmission.
3. *Normalized routing overhead* (*packets*): The number of routing control packets per data packet delivered at the destination.
4. *Average path length*: The average length of the paths discovered by the protocol. It is calculated by averaging the number of hops taken by each data packet to reach the destination.
5. *Average route acquisition latency*: The average delay between the sending of an RREQ packet by a source for discovering a route to a destination and the receipt of the first corresponding RREP.
6. *Average end-to-end delay of transferred data packets*: This includes all possible delays caused by buffering during route discovery, queuing at the interface-queue, retransmission delays at the medium access control layer, and propagation and transfer times.

### 3.9.1 STEP vs. AODV without Attackers

In this subsection, the network topology consists of 100 randomly placed over a 1500×1500 $m^2$ flat-grid, and each simulation run takes 300 simulated seconds. Figure 3.13 shows that the packet delivery ratio among STEP, ESTEP, and AODV are very close and above 90 % in all scenarios. This suggests that both STEP and ESTEP are effective in discovering and maintaining routes for delivery of data packets even in relatively high mobility scenarios. Figure 3.14 shows that STEP's routing overhead is higher than that of AODV in terms of bytes. That is due to the increase in size of routing control packets with digital signatures in STEP. As mobility increases, the amount of control overhead of STEP increases linearly. On the other hand, ESTEP remains efficient, as compared to STEP, by avoiding multiple signatures in RREQ broadcast messages. Figure 3.15 shows the average number of control packets transmitted per data packet delivered. The three routing protocols demonstrate nearly the same amount of routing overhead. AODV has the advantage of smaller control packets; smaller packets have a higher probability of successful reception at the destination.

Figure 3.16 shows that AODV selects a slightly longer path when compared to STEP and ESTEP. Since all three protocols find the shortest path explicitly based on the same route discovery mechanism of AODV, the length of paths should not differ significantly. However, it is possible that due to a higher contention or queuing delay along the shortest path, a sub-optimal path is being used instead. In STEP, due to the additional delay of signature verification and generation in each hop, the possibility of finding a longer path instead of the shortest one decreases as compared to AODV. Figure 3.17 shows that the average route acquisition latency for STEP is approximately double that for AODV. This is due to the additional processing delay for multiple digital signature generations and verifications at each node for control packets. Since multiple digital signatures are not used for RREQ messages in ESTEP, ESTEP can find a route with a smaller additional delay when compared to STEP. Figure 3.18 shows that the average end-to-end delay for both STEP and ESTEP are slightly higher than that of AODV. Although STEP

and ESTEP have higher average route acquisition delay than AODV, the number of route discoveries performed is a small fraction compared with the number of data packets delivered. Therefore, the effect of the route acquisition latency on the average end-to-end delay of data packets is not significant. Note that the processing delay of data packets is identical in all three protocols.

### 3.9.2 SeAODV, SAODV, and AODV without Attackers

We first compare among AODV, SAODV, and SeAODV without attackers. Figure 3.19 shows that the packet delivery fractions of all protocols are very close. This suggests that both SAODV and SeAODV are effective in discovering and maintaining routes for delivery of data packets, even with relatively high node mobility.

Figure 3.20 shows that the routing overhead of both SAODV and SeAODV are higher than that of AODV. This is obviously due to the increase in size of routing control packets in both SAODV and SeAODV with digital signatures.

Figure 3.21 shows the average number of control packets transmitted per data packet delivered. Basically, all three protocols demonstrate nearly the same amount of routing packet overhead due to no attackers. AODV has the advantage of small control packets; smaller packets have a higher probability of successful reception at the destination. However, due to the IEEE 802.11 DCF medium access control for unicast transmissions [i.e., RTS (Request-To-Send) and CTS (Clear-To-Send) messages], a significant part of time is spent in acquiring the channel not in transmission.

Figure 3.22 shows that both SeAODV and SAODV select a slightly longer path when compared to AODV. Since both protocols find the shortest path explicitly based on the route discovery mechanism of AODV, the length of paths should not differ significantly. However, due to the secure route maintenance mechanism with the suspicious value, any link breakage in the path will be regarded as a dropping attack in SeAODV. Therefore, sometimes, the sub-optimal

path is being used instead of the shortest path.

Figure 3.23 shows that the average route acquisition latency for SeAODV is approximately double that for AODV and 20 % more than that of SAODV. This is due to the additional processing delay for digital signature generation and verification for control packets, especially double signature for an RREP. Moreover, since only destination can send an RREP to source node in both SeAODV and SAODV, the route acquisition latency increases.

Figure 3.24 shows that the average end-to-end delay for both SeAODV and SAODV is up to 15 % higher than that of AODV. Note that the number of route discoveries performed is a small fraction of the number of data packets delivered. Hence, the effect of the route acquisition latency on the average end-to-end delay of data packets is not significant.

In the simulation model, the random waypoint model is used for user's mobility. The two variables in the random waypoint model are the speed and pause time. In Figures 3.19 – 3.24, we show the performance comparisons under different pause time. Although in the thesis we did not include the results by varying the speed, the preliminary results showed that the relative performance between AODV, SAODV, and SeAODV remain the same when the speed is varied.

### 3.9.3 SeAODV, SAODV, and AODV with Blackhole Attackers

Figures 3.25–3.30 show the performance metrics as a function of the number of black hole attackers in the network. We assume that black hole attackers only drop data packets but not routing packets. For both AODV and SAODV, the network will not be able to detect the presence of the malicious attacker as shown in Figure 3.11. Therefore, malicious attackers can drop data messages in a broken link. In Figure 3.25, SeAODV maintains a higher packet delivery fraction than both AODV and SAODV. It shows our proposed mechanism can effectively isolate malicious black hole attackers by using SDF proposed in Section 3.8.1.

Figures 3.26–3.27 show that SeAODV's both the byte and packet routing overheads are significantly higher and increase exponentially as the number of attackers increases. This is due

to the fact that SeAODV detects the black hole attackers and initiates a new route discovery to avoid those nodes. The increase of the routing control overhead is due to the increase of the number of RREQ, RREP, and RERR control packets in the network.

Figure 3.28 indicates that the average path lengths of SAODV are very similar to that of AODV. On the other hand, the length of SeAODV is almost constant. This explains that blackhole attackers effectively prevents longer path from being used to deliver data messages from source to destination. If the attacker is near the initiator of route discovery, this attack can even prevent routes more than two hops long from being used.

Figure 3.29 shows the average route acquisition delays for both SAODV and SeAODV are much higher than that of AODV. This is due to the security data for processing SeAODV's routing control messages as like Figure 3.23.

Results in Figure 3.30 indicate that both AODV and SAODV have slightly lower average end-to-end delay when compared with SeAODV. In general, SeAODV induces additional control messages after detecting malicious black hole attackers to the network. Thus, the average end-to-end delay for data message increases. However, note that the average end-to-end delay of both AODV and SAODV decreases slightly as the number of black hole attackers increases. The rationale is that black hole attackers drop data message at the intermediate nodes. Since the dropped packets are not counted in the end-to-end delay calculation, the average end-to-end delay decreases

### 3.9.4 SeAODV vs. AODV with Routing Table Tampering Attackers

Figures 3.31–3.33 show the performance metrics as a function of the number of routing table tampering attackers in the network. In this scenario, whenever the routing table tampering attacker in node *A*, for example, receives an RREQ from its neighbor *B*, node *A* can redirect traffic towards itself by unicast to node *B* an RREP containing a higher *destination sequence number* for the destination than the value last advertised. Node *A* will then become one of the

intermediate nodes of the path. Moreover, in node *A*'s routing table module, the attacker alters the next-hop address field of the corresponding destination entry to an unreachable or non-existing address. As a result, data packets passing through node *A* will never reach their intended destination. In this sub-section, we allow an intermediate node to generate an RREP if it has a fresh enough route to satisfy the request. We also refer to SeAODV with TRM as SeAODV-T.

Results in Figure 3.31 indicate that when the number of routing table tampering attackers increases, the difference of the packet delivery fraction between SeAODV-T and AODV becomes significant. Figure 3.32 shows that AODV has lower average end-to-end delay than SeAODV-T. This is because routing and data packets dropped by table tampering attackers at intermediate nodes are not counted in the end-to-end delay calculation, and it reduces the level of congestion in the network. Figure 3.33 shows that the routing overhead in AODV increases as the number of routing table tampering attackers increases. As the routes become unstable due to the routing table tampering attackers, more routing packets are generated in AODV.

## 3.10 Summary

In this chapter, we proposed both the Tamper Resistant Module (TRM) and the Secure Table Entry Protection (STEP) mechanism to prevent routing table tampering attacks. STEP provides the authentication for both the destination sequence number and hop-count fields in the routing table entry. The receiving node can confirm the correctness of message by verifying the signatures of two consecutive upstream nodes and the source node, which creates the packet. We described how STEP can be incorporated in the AODV routing protocol. We proposed the Efficient STEP (ESTEP), which can avoid the use of expensive multiple digital signatures on RREQ broadcast messages. We have proposed a secure routing extension (SeAODV) and secure data forwarding (SDF) mechanism for the AODV routing protocol. A digital signature is used for both RREQ and RREP to prevent malicious users from redirect traffic toward itself. Both RREQ and RERR are modified to detect and avoid data dropping attackers. For secure data transmission,

we have proposed the use of HMAC to maintain the message integrity. Simulation results showed that both STEP and ESTEP continue to maintain a high packet delivery fraction and a small end-to-end delay at the expense of slightly higher route acquisition latency and control overhead in route discovery. In the presence of data packet dropping attackers, SeAODV continues to maintain a high packet delivery fraction and a small end-to-end delay.

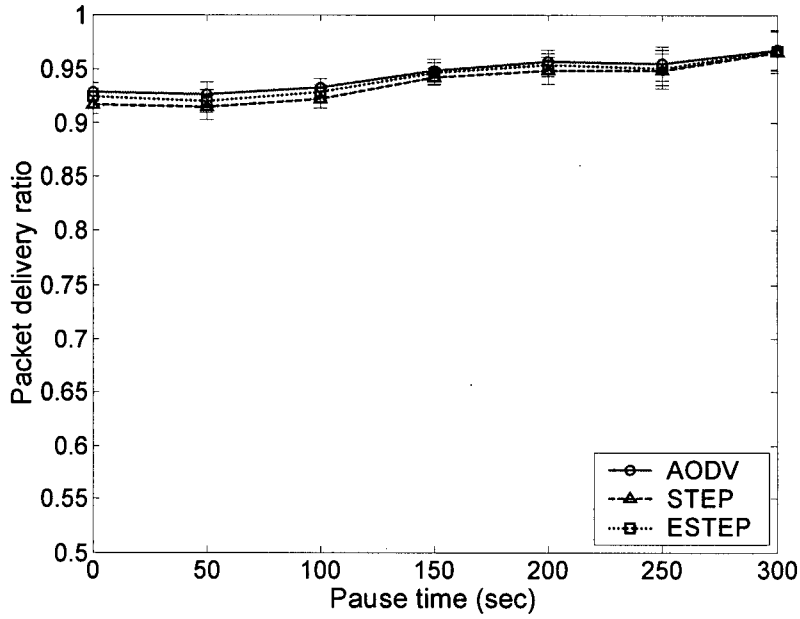



Figure 3.13 Packet delivery ratio among STEP, ESTEP, and AODV over a range of pause time without attackers.

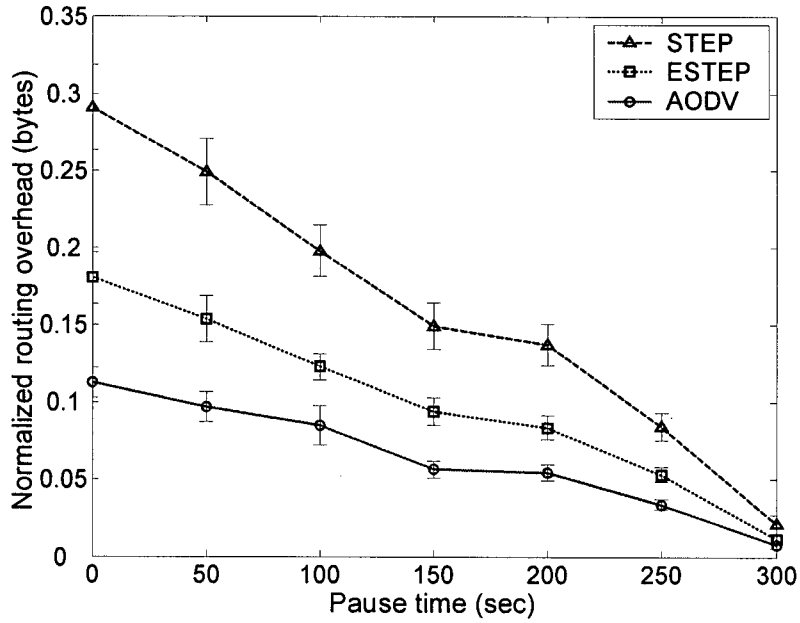



Figure 3.14 Normalized byte routing overhead among STEP, ESTEP, and AODV over a range of pause time without attackers.

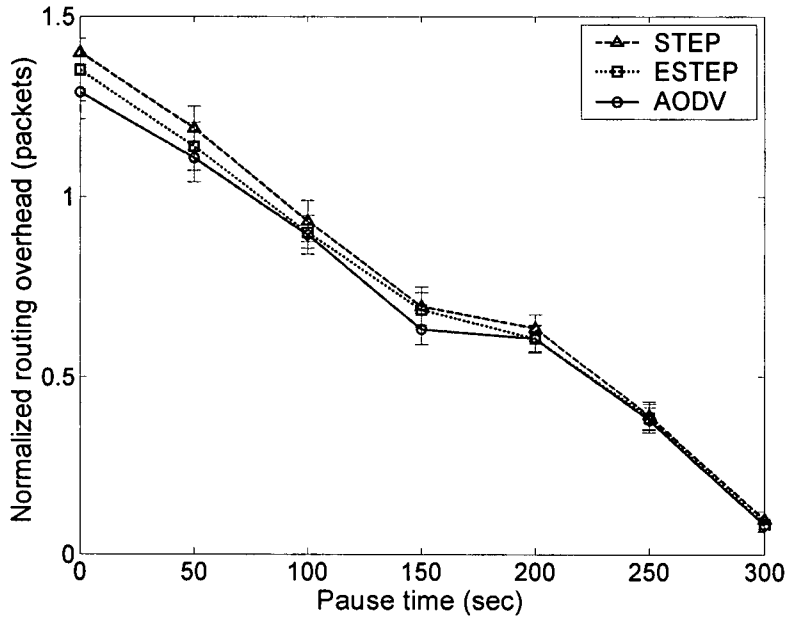



Figure 3.15 Normalized packet routing overhead among STEP, ESTEP, and AODV over a range of pause time without attackers.

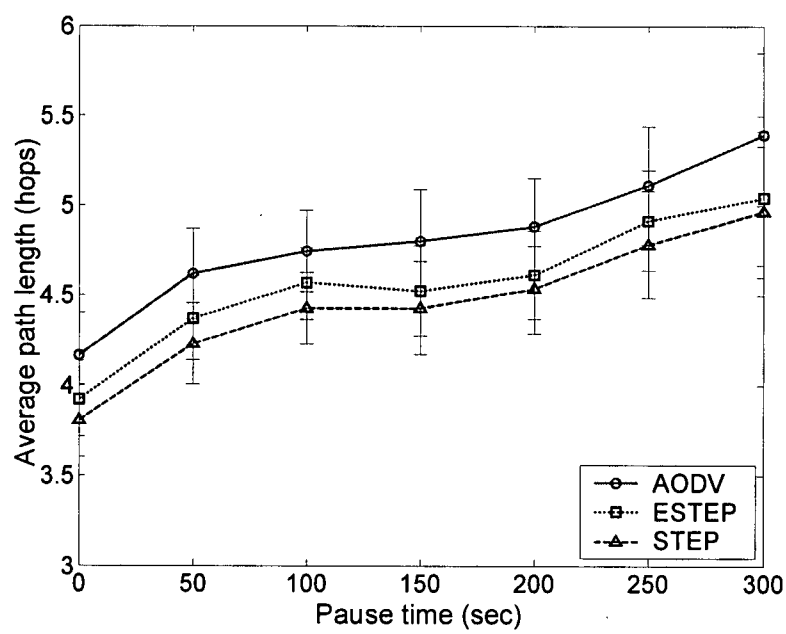



Figure 3.16 Average path length among STEP, ESTEP, and AODV over a range of pause time without attackers.

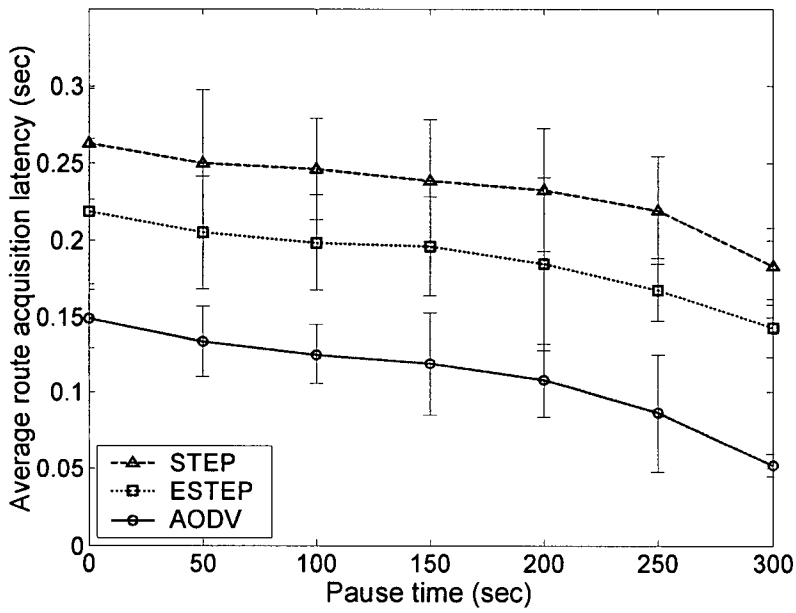



Figure 3.17 Average route acquisition latency among STEP, ESTEP, and AODV over a range of pause time without attackers.

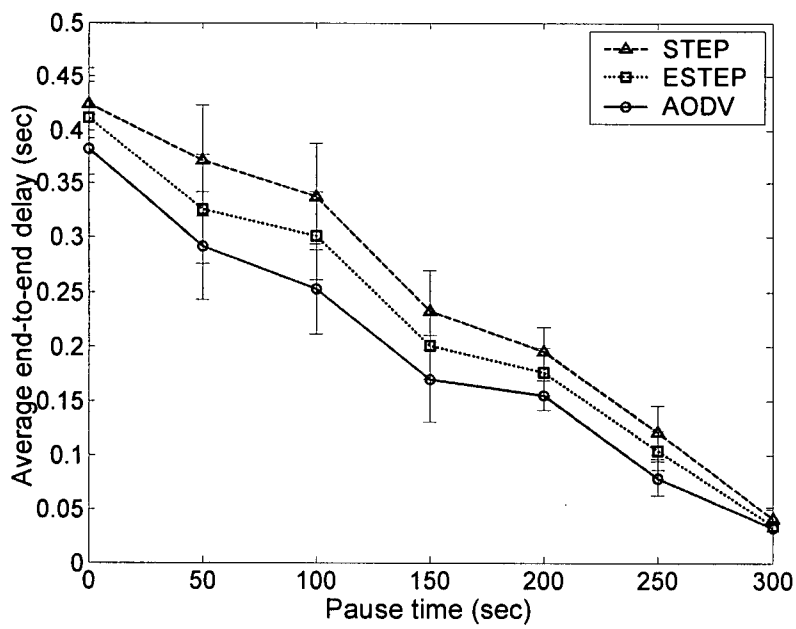



Figure 3.18 Average end-to-end delay among STEP, ESTEP, and AODV over a range of pause time without attackers.

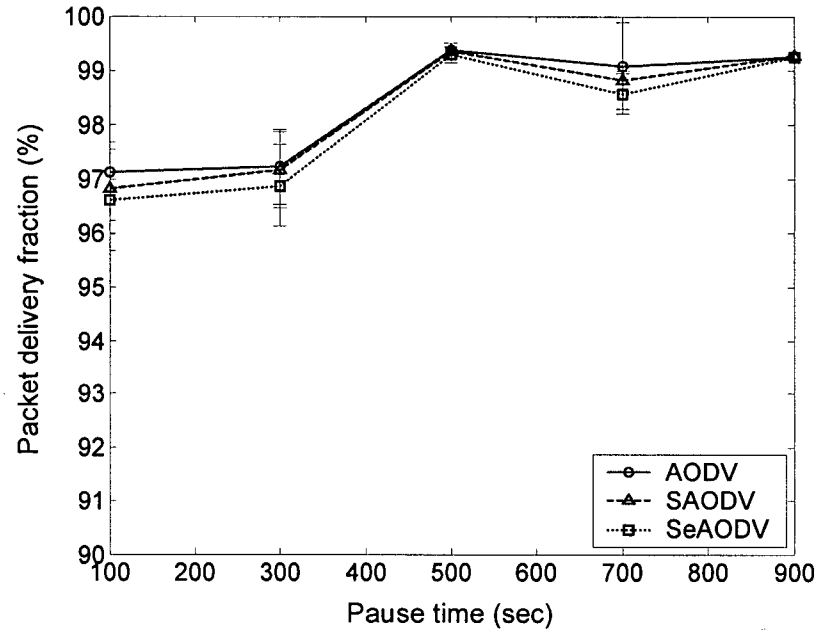



Figure 3.19 Packet delivery fraction among SeAODV, SAODV, and AODV over a range of pause time without attackers.

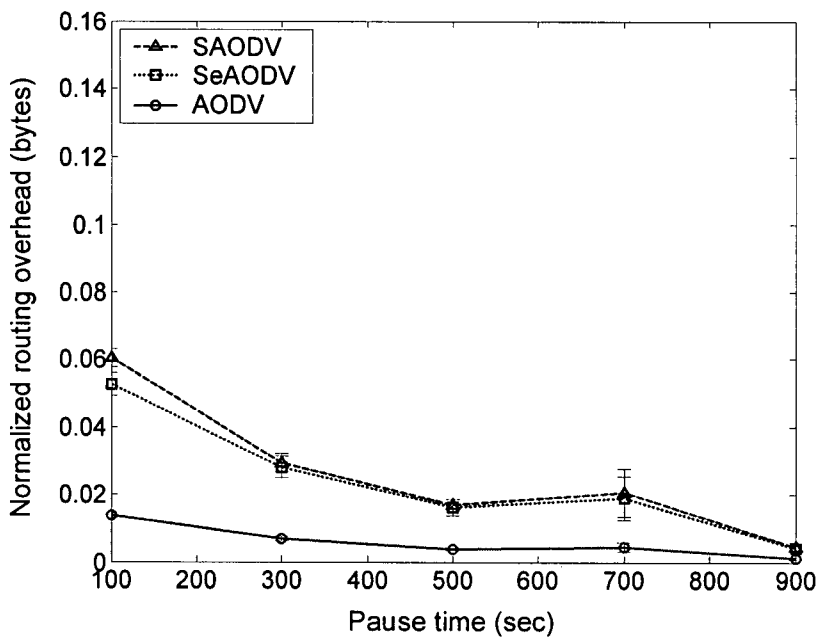



Figure 3.20 Normalized byte routing overhead among SeAODV, SAODV, and AODV over a range of pause time without attackers.

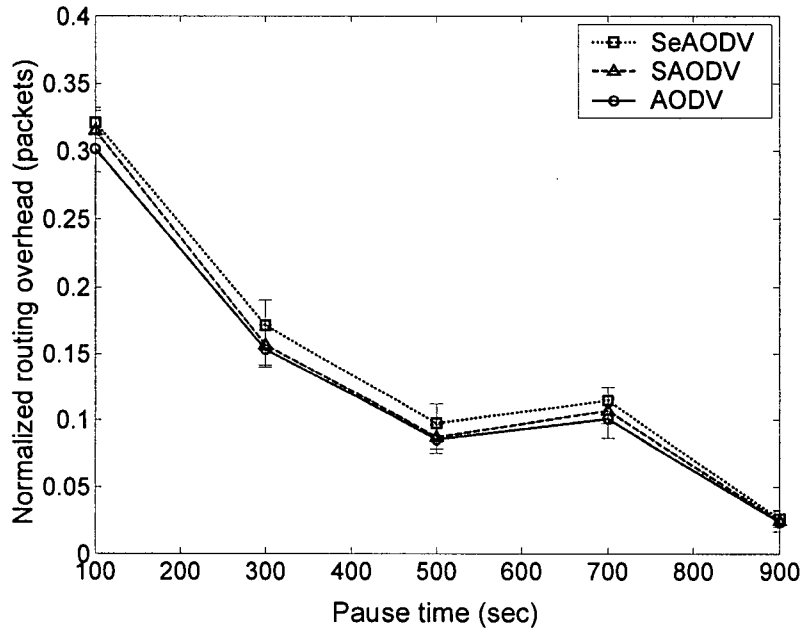



Figure 3.21 Normalized packet routing overhead among SeAODV, SAODV, and AODV over a range of pause time without attackers.

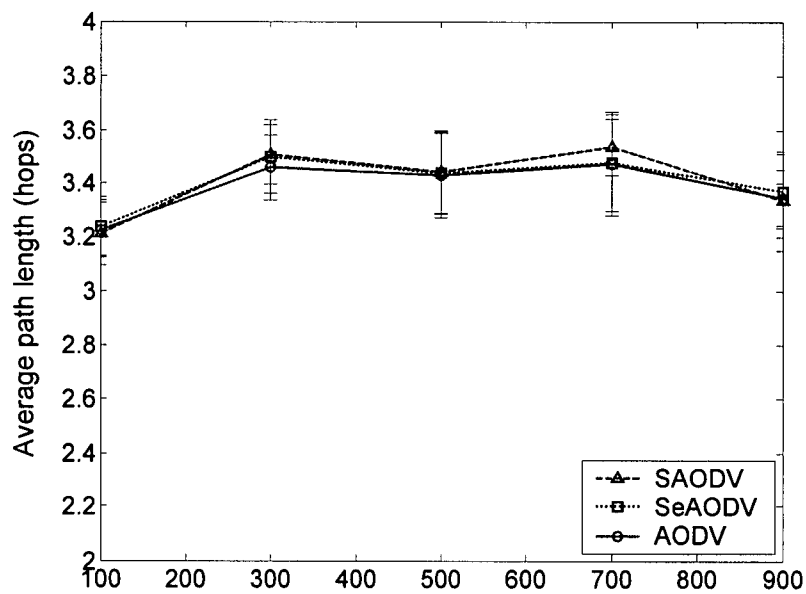



Figure 3.22 Average path length among SeAODV, SAODV, and AODV over a range of pause time without attackers.

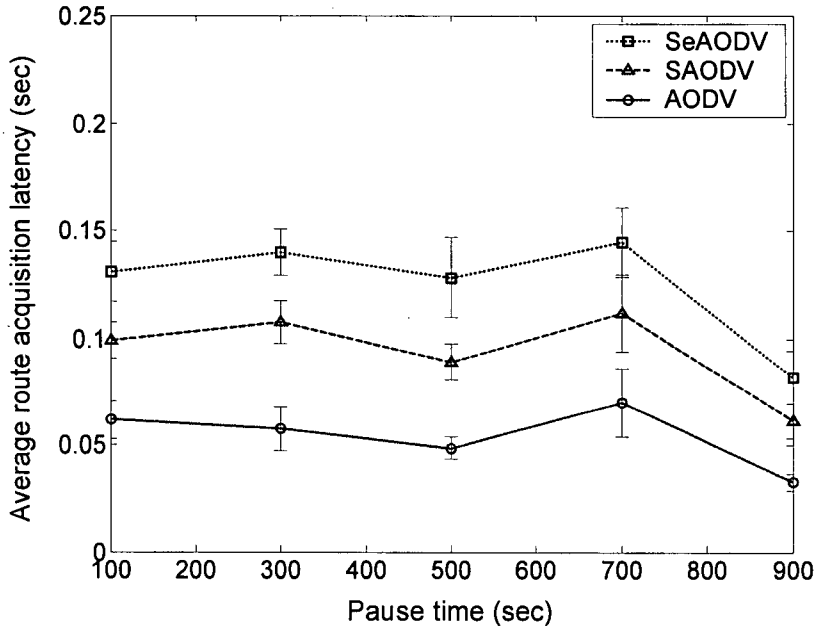



Figure 3.23 Average route acquisition latency among SeAODV, SAODV, and AODV over a range of pause time without attackers.

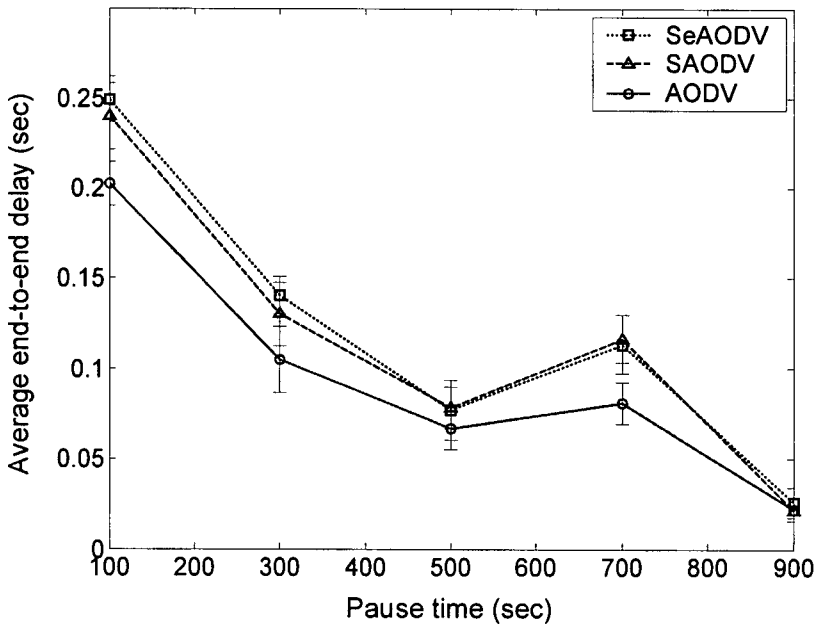



Figure 3.24 Average end-to-end delay among SeAODV, SAODV, and AODV over a range of pause time without attackers.

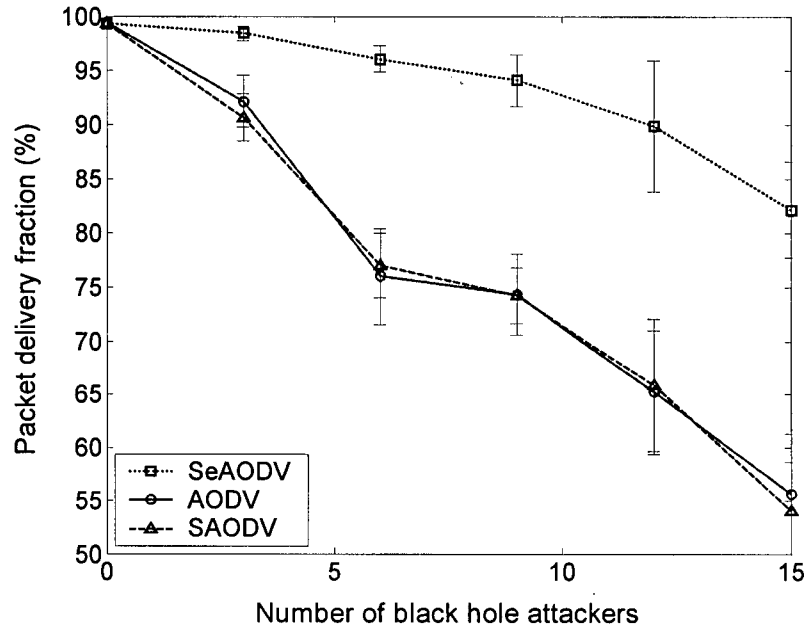



Figure 3.25 Packet delivery fraction among SeAODV, SAODV, and AODV with varying number of data black hole attackers (pause time = 500 sec).

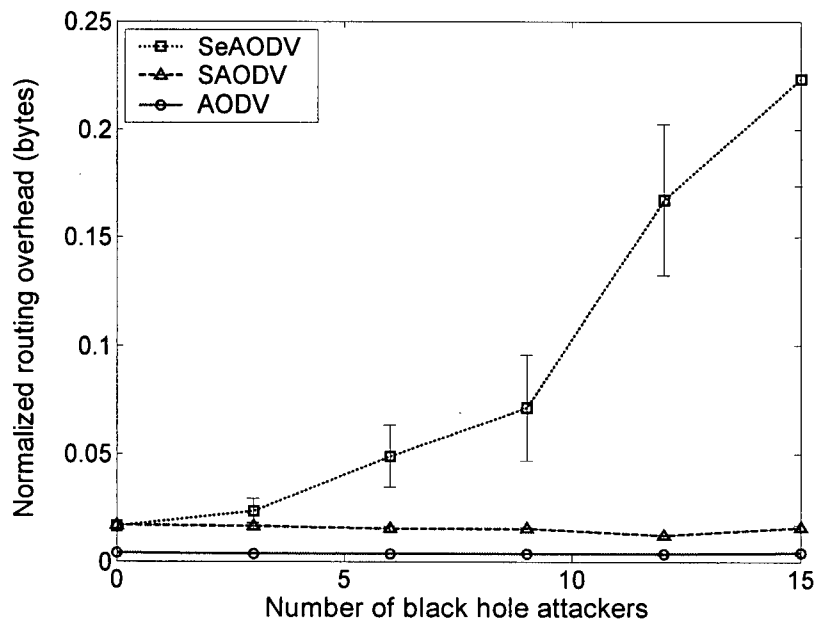



Figure 3.26 Normalized byte routing overhead among SeAODV, SAODV, and AODV with varying number of data black hole attackers (pause time = 500 sec).

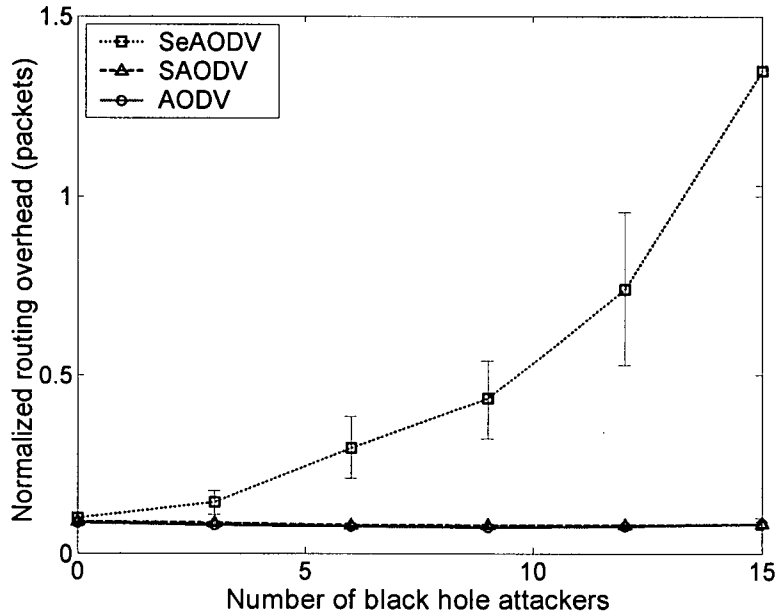



Figure 3.27 Normalized packet routing overhead among SeAODV, SAODV, and AODV with varying number of data black hole attackers (pause time = 500 sec).

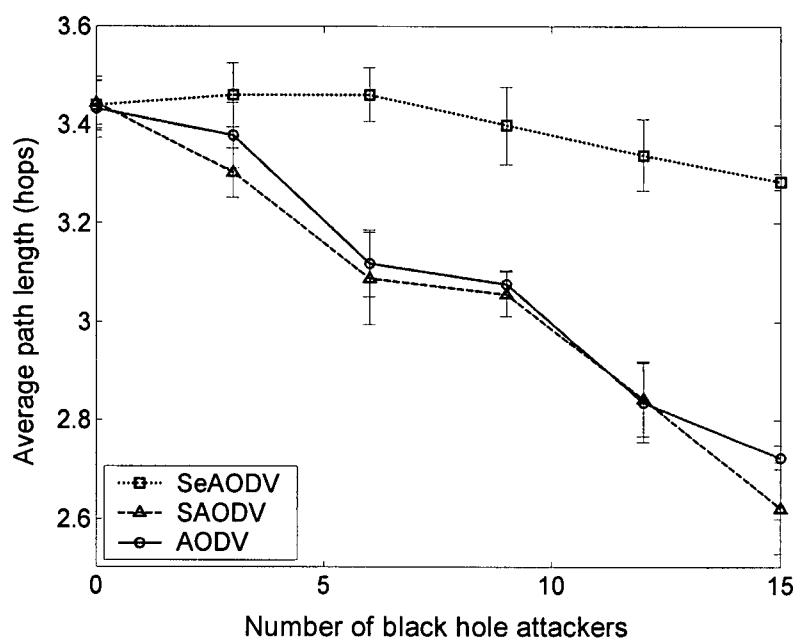



Figure 3.28 Average path length among SeAODV, SAODV, and AODV with varying number of data black hole attackers (pause time = 500 sec).

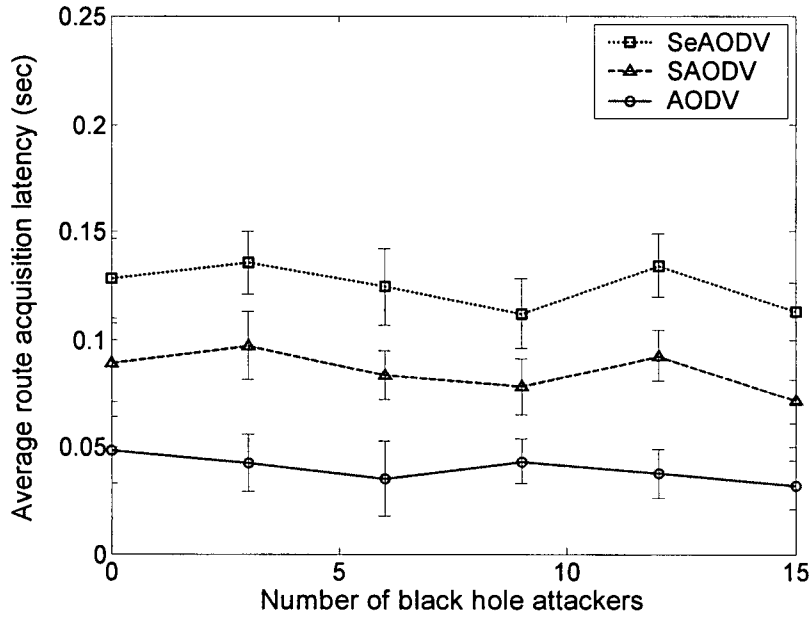



Figure 3.29 Average route acquisition latency among SeAODV, SAODV, and AODV with varying number of data black hole attackers (pause time = 500 sec).

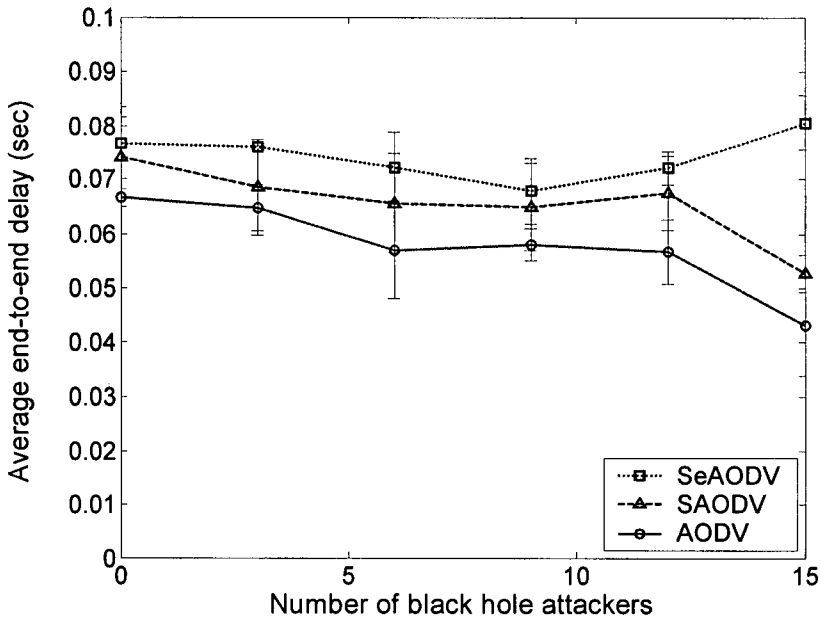



Figure 3.30 Average end-to-end delay among SeAODV, SAODV, and AODV with varying number of data black hole attackers (pause time = 500 sec).

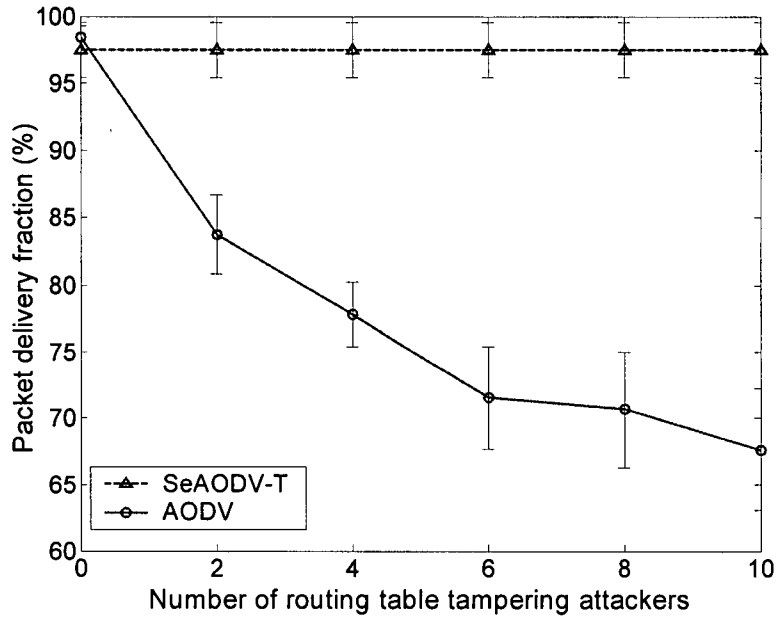



Figure 3.31 Packet delivery fraction between SeAODV-T and AODV with routing table tampering attackers in the network (500 sec. pause time).

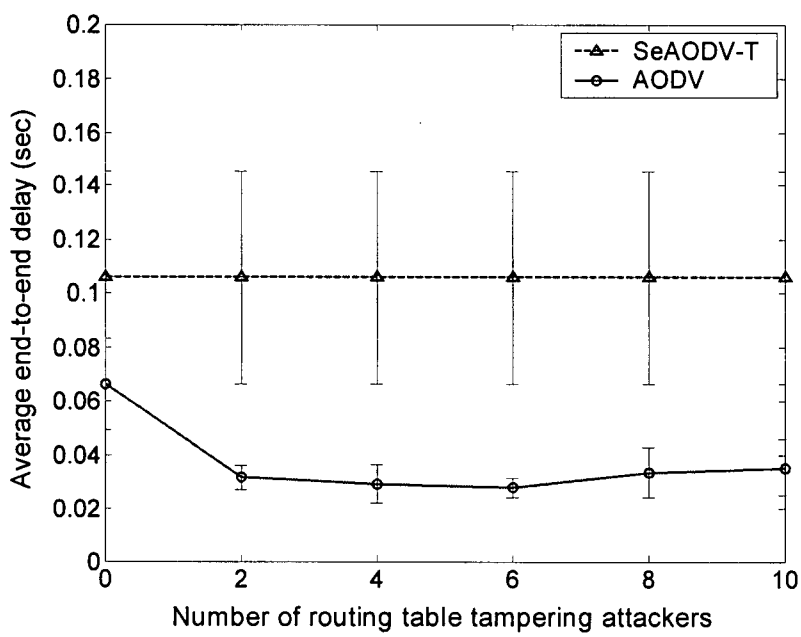



Figure 3.32 Average end-to-end delay between SeAODV-T and AODV with routing table tampering attackers in the network (500 sec. pause time).

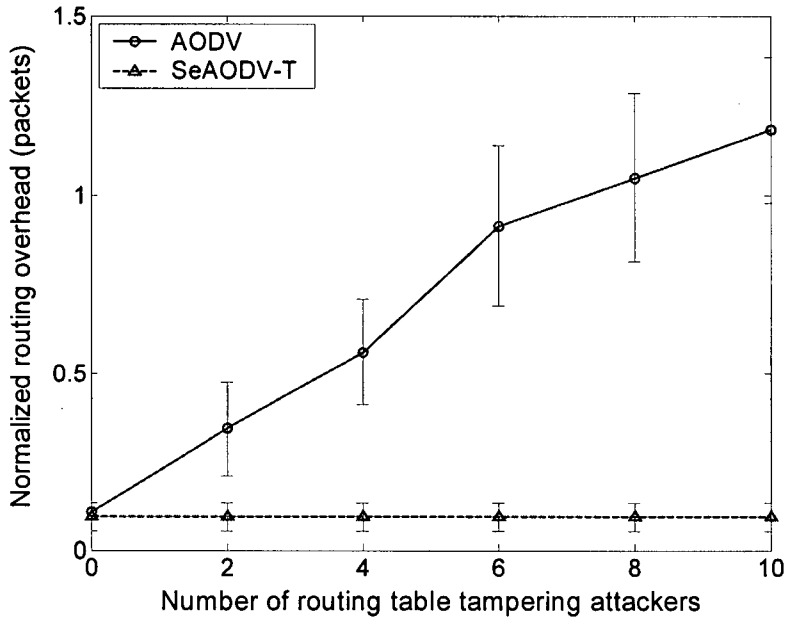



Figure 3.33 Average end-to-end delay between SeAODV-T and AODV with routing table tampering attackers in the network (500 sec. pause time).

# Bibliography


[1] J.-H. Song, W.S.V. Wong, V.C.M. Leung, and Y. Kawamoto, "Secure routing with tamper resistant module for mobile ad hoc networks," *ACM Mobile Computing and Communications Review*, vol. 7, issue 3, July 2003.

[2] J.-H. Song, W.S.V. Wong, V.C.M. Leung, and Y. Kawamoto, "Secure AODV routing protocol with table entry protection for mobile ad hoc networks," submitted to the *IEEE Journal on Selected Areas in Communications*.

[3] C.E. Perkins, E. Belding-Royer, and S.R. Das, "Ad hoc On-demand Distance Vector (AODV) routing," *IETF RFC* 3561, July 2003.

[4] M. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proc. of ACM Workshop on Wireless Security*, Atlanta, GA, Sept. 2002.

[5] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: keyed-hashing for message authentication," *IETF RFC* 2104, Feb. 1997.

[6] C. Madson and R. Glenn, "The use of HMAC-SHA-1-96 within ESP and AH," *IETF RFC* 2404, Nov. 1998.

[7] R. Rivest, "The MD5 message-digest algorithm," *IETF RFC* 1321, Apr. 1992.

[8] A. C-F. Chan, "Distributed symmetric key management for mobile ad hoc networks," in *Proc. of IEEE Infocom*, Hong Kong, Mar. 2004.

[9] D.B. Johnson, "ECC, future resiliency and high security systems," *Certicom White Chapter*, Mar. 1999.

[10] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*, John Wiley & Sons, Inc., 1996.

[11] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," in *Proc. of ACM MobiCom*, Atlanta, GA, Sept. 2002.

[12] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, and E.M. Belding-Royer, "A secure routing protocol for ad hoc networks," in *Proc. of International Conference on Network Protocols*, Paris, France, Nov. 2002.

[13] P. Papadimitratos and Z.J. Haas, "Secure routing for mobile ad hoc networks," in *Proc. of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Antonio, TX, Jan. 2002.

[14] Y.-C. Hu, A. Perrig, and D.B. Johnson, "SEAD: secure efficient distance vector routing for mobile wireless ad hoc networks," in *Proc. of IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[15] P. Papadimitratos and Z.J. Haas, "Secure link state routing for mobile ad hoc networks," in *Proc. of IEEE Workshop on Security and Assurance in Ad hoc Networks*, Orlando, FL, Jan. 2003.

[16] D.B. Johnson, D.A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," *IETF Internet Draft* (work in progress), July 2004.

[17] A. Perrig, R. Canetti, D. Song, D. Tygar, and B. Briscoe, "TESLA: multicast source authentication transform introduction," *IETF Internet Draft of Multicast Security Working Group* (work in progress), Aug. 2004.

[18] Trimble Navigation Limited. *Datasheet & specifications for Trimble Thunderbolt GPS Disciplined Clock.* Sunnyvale, CA. Available at http://trl.trimble.com/ docushare/ dsweb/Get/Document-10015/

[19] P. Ning and K. Sun, "How to misuse AODV: a case study of insider attacks against mobile ad hoc routing protocols," in *Proc. of IEEE Information Assurance Workshop*, West Point, NY, June 2003.

[20] W. Wang, Y. Lu, and B. Bhargava, "On vulnerability and protection of ad hoc on-demand distance vector protocol," in *Proc. of the International Conference on Telecommunication*, Papeete, France, Feb./Mar. 2003.

[21] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless network," in *Proc. of IEEE Infocom*, San Francisco, CA, Mar./Apr. 2003.

[22] L. Zhou and Z.J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, issue 6, Nov./Dec. 1999.

[23] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proc. of ACM MobiHoc*, Long Beach, CA, Oct. 2001.

[24] M. Brown, D. Cheung, D. Hankerson, J.L. Hernandez, M. Kirkup, and A. Menezes, "PGP in constrained wireless devices," in *Proc. of the USENIX Security Symposium*, Denver, CO, Aug. 2000.

[25] J.R. Nickerson, S.T. Chow, H.J. Johnson, and Y. Gu, "Tamper resistant software: extending trust into a hostile environment," in *Proc. of ACM Multimedia and Security Workshop*, Ottawa, ON, Canada, Oct. 2001.

[26] O. Kömmerling and M.G. Kuhn, "Designing principles for tamper-resistant smartcard processors," in *Proc. of the USENIX workshop on Smartcard Technology*, Chicago, IL, May 1999.

[27] M. Mambo, T. Murayama, and E. Okamoto, "A tentative approach to constructing tamper-

resistant software," in *Proc. of ACM New Security Paradigms Workshop*, Langdale, Cumbria, UK, 1997.

[28] D. Aucsmith, "Tamper resistant software: an implementation," in *Proc. of the International Information Hiding Workshop*, London, UK, May/June 1996.

[29] I.D. Chakeres and E.M. Belding-Royer, "AODV routing protocol implementation design," in *Proc. of the International Workshop on Wireless Ad Hoc Networking*, Tokyo, Japan, Mar. 2004.

[30] V. Kawadia, Y. Zhang, and B. Gupta, "System services for ad-hoc routing: architecture, implementation and experiences," in *Proc. of ACM Mobisys*, San Francisco, CA, May 2003.

[31] J. Kadlecsik, H. Welte, J. Morris, M. Boucher, and R. Russel, The netfilter/iptables Project. Available at http://www.netfilter.org/

[32] L.B. Michael, M.J. Mihaljevic, S. Haruyama, and Ryuji Kohno, "A framework for secure download for software-defined radio," *IEEE Communications Magazine*, vol. 40, no. 7, pp. 88-96, July 2002.

[33] R. Anderson and M. Kuhn, "Tamper resistance–a cautionary note," in *Proc. of the USENIX Electronic Commerce Workshop*, Oakland, CA, Nov. 1996.

[34] M.K. Marina and S.R. Das, "On-demand multipath distance vector routing in ad hoc network," in *Proc. of the International Conference for Network Protocols*, Riverside, CA, Nov. 2001.

[35] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of ACM MobiCom*, Boston, MA, Aug. 2000.

[36] W. Diffie, P. van Oorschot, and M. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes, and Cryptography*, vol. 2, issue 2, pp. 107-125, June 1992.

[37] The network simulator - NS-2 notes and documentation and source code. Available at http://www.isi.edu/nsnam/ns/

# Chapter 4 Secure Position-based Routing Protocol for Mobile Ad hoc Networks[4]

## 4.1 Introduction

In this chapter, we provide security mechanisms for both data and control messages in position-based routing protocols [1]. We analyze the security threats and identify the security requirements for position-based routing protocols in MANETs. In consideration of these requirements, we design a Secure Geographic Forwarding (SGF) mechanism, which provides *source authentication*, *neighbor authentication*, and *message integrity* by using both the shared key and the broadcast authentication [2] protocol. Combining SGF with the Grid Location Service (GLS) [3], we design a Secure GLS (SGLS) where any receiver can verify the correctness of location messages. To detect and isolate *message tampering* and *dropping* attackers, the Local Reputation System (LRS) is integrated with GLS. We present simulation results to show that in the presence of *message dropping* attackers, GLS with LRS continues to maintain a high message delivery ratio at the expense of a slightly higher average end-to-end delay and routing overhead when compared to the GLS without LRS. In addition, results show that SGLS remains efficient by using efficient cryptographic mechanisms.

This chapter is organized as follows. Sections 4.2–4.4 give overviews on both the Timed Efficient Stream Loss-tolerant Authentication (TESLA) [4] and the TESLA with Instant Key disclosure (TIK) [2], position-based routing protocols, and reputation systems, respectively. In Section 4.5, we analyze the possible attacks and describe the security requirements for position-based routing protocols in MANETs. Section 4.6 explains the rationale of the

[4] Manuscript to be submitted to a journal. A shorter version of this manuscript has been published in *Proc. of ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Network*, Venice, Italy, Oct. 2004.

assumptions we make in our framework. Our proposed Secure Geographic Forwarding (SGF) mechanism is described in Section 4.7. In Section 4.8, we describe our proposed Secure Grid Location Service (SGLS). The integration of Local Reputation System (LRS) into GLS is described in Section 4.9. The performance comparisons among the original GLS, SGLS, and LRS are presented in Section 4.10. Summary is given in Section 4.11.

## 4.2 TESLA & TIK

By applying one-way function repeatedly on the initial secret value, we can generate a one-way key chain. This chain can be used to authenticate broadcast messages [4]. A broadcast authentication protocol enables the receivers to verify that the broadcast messages received were actually sent by the claimed sender. Although there are several proposals for secure broadcast authentication, it is proven that one cannot build an efficient collusion resistant authentication protocol without relying on digital signatures or time synchronization [5].

TESLA [4] applies the MAC to a message for broadcast authentication. It provides key secrecy by using clock synchronization and delayed key disclosure. Each node chooses a random initial secret key $K_n$ and generates a one-way key chain by repeatedly computing a one-way hash function $h$ [6][7] as shown in Figure 4.1. A node can compute any previous key $K_j$ from a key $K_i$ where $j < i$, by $K_j = h^{i-j}[K_i]$. To authenticate any received value on the one-way chain, a node applies the above hash function to the received value to determine if the computed value matches a previously known authentic public key $K_0$ on the chain. Each node pre-determines a schedule at which it discloses each key of its one-way key chain, in the reverse order from generation.

In TIK [2], the sender can disclose the key within the same message when all nodes have tightly synchronized clocks (e.g., within 200 ns) to defend against *wormhole* attack. This level of time synchronization can be achieved via some off-the-shelf GPS devices [8]. To schedule the key disclosure time within a message's transmission, a minimum payload length is determined according to the transmission rate of the physical layer. When the IEEE 802.11 DCF

[9] is used, the minimum message size can be reduced by piggybacking the MAC in the RTS frame.

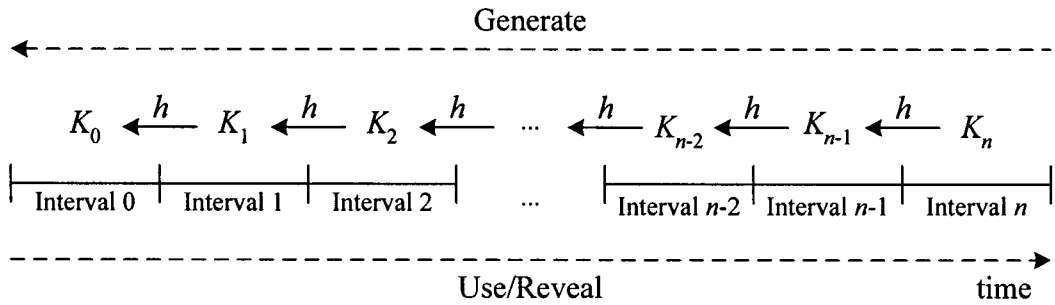



Figure 4.1 Example of one-way chain.

## 4.3 Position-based Ad hoc Routing Protocols

In position-based routing protocol, the task of routing packets from a source to a destination can be separated into two distinct aspects [10]:

- Discovery of the position of the destination;
- The actual forwarding of packets based on the position information.

The routing decision at each node is based on the location information of its forwarding neighbor and the destination. In order to learn the current location of a destination, the use of a *location service* is necessary. In Sections 4.3.1–4.3.4, we classify the existing location services based on how to decide the location server and how many nodes host the location service. After deciding the location of destination by using the location service, all messages can be forwarded by using different packet forwarding strategies. In Sections 4.3.5 and 4.3.6, we describe two main packet forwarding strategies: unicast forwarding and directional flooding.

### 4.3.1 Grid Location Service (GLS)

The Grid Location Service (GLS) [3] is a distributed location service which calls for nodes to

maintain location of specific subsets of the nodes based on the node's identifier (ID) as shown in Figure 4.2. GLS divides the area that contains a MANET into a hierarchy of squares. Four order-$n$ squares make up an order-($n$+1) square, and so on. Each node periodically broadcasts a list of all neighbors using a HELLO message. Therefore, each node can maintain a table of immediate neighbors as well as each neighbor's neighbors. Each entry in the table includes the node's *unique ID*, *location*, *speed*, and a *timestamp*. Each node recruits nodes with IDs "close" to its own ID to serve as its location servers. Note that the ID space is considered circular. Consider Figure 4.2 again as an example. To perform a location discovery, node $S$ sends a location query message using greedy geographic forwarding [11] to the node with the least ID greater than node $D$ for which $S$ has the location information (i.e., LS1). If LS1 does not have the location information, it will then forward the message to another location server LS2. If LS2 has the location information of node $D$, it will send the location query message to node $D$.

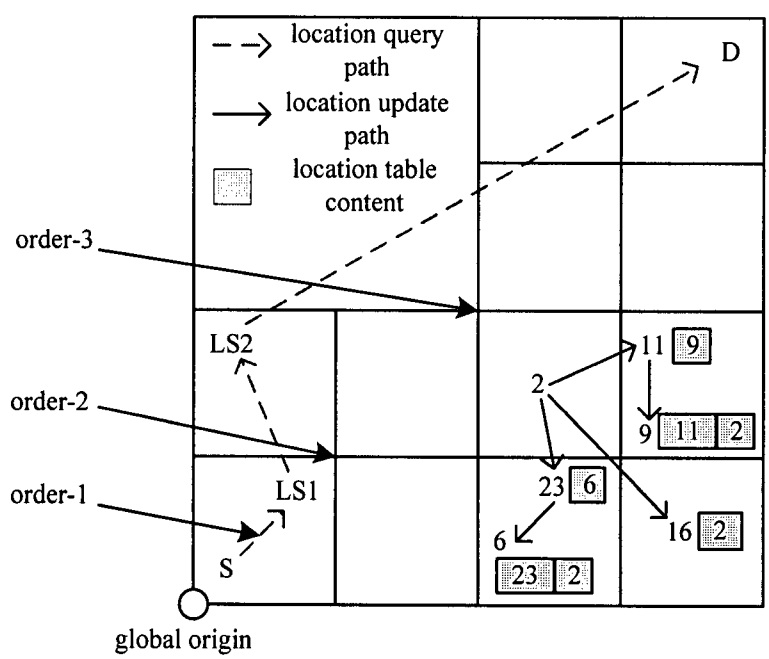



Figure 4.2 Location update and query in GLS.

### 4.3.2 Virtual Home Region (VHR)

The Virtual Home Region (VHR) [12] is a location service based on the Location-Dependent Address (LDA). The main objective of LDA management is to distribute the location information inside the network in a dynamic and scalable manner. The main functions of VHR are: (1) maintaining location information; and (2) distributing location information inside the network. Each node advertises its current LDA (location) to a geographic region called the VHR. The VHR has a fixed center $C_{VHR}$ and a variable radius that adapts to the density of the area containing the VHR, in order to maintain an approximately constant number of nodes inside the VHR. The fixed center is computed using some predefined known hash function $h$. The function $h$ is a static mapping between the space of End-system Unique Identifiers (EUIs) and the geographic space of a network. Each node does location update by sending the location update message toward its own VHR. All nodes in $A$'s VHR store the mapping between $A$'s EUI and LDA.

### 4.3.3 Distance Routing Effect Algorithm for Mobility (DREAM)

In the Distance Routing Effect Algorithm for Mobility (DREAM) [13] location service, each node maintains a table containing the location information of all the nodes in the network and updates the location information in a promiscuous manner. Each node floods the location update message adaptively based on its mobility and relative distance to other nodes. The frequency of location update increases when either the speed increases or the location of nodes becomes closer.

### 4.3.4 Quorum based Location Service

Quorum based location service is proposed in [14]. Uniform quorum systems are used to provide a distributed location management scheme. Node location information is maintained in location databases that form a virtual backbone (i.e., quorum). Initially, the virtual backbone is constructed using a non-position-based routing protocol such as flooding. Each node sends the

location update messages to the nearest backbone node, which chooses a quorum of backbone nodes to host this location information. When a node requires the location information, it sends a location query message to the nearest backbone node, which in turn contacts the nodes of a quorum. Since the intersection of two quorums is non-empty, the querying node can receive at least one response with the current location information.

### 4.3.5 Unicast Forwarding

A unicast forwarding scheme decides the next node based on the information about the location of the current node, its neighbors, and the final destination. Each intermediate node applies this rule until the destination is reached. There are many unicast forwarding algorithms based on the optimization criterion applied in each forwarding steps. For example, the widely used greedy forwarding strategy [11] forwards message to a neighbor that is closest to the destination in terms of the Euclidean distance. Compass routing [15] scheme considers the deviation (angle between next hop, current, and destination node) from the line connecting the current sender and destination. When each node can adapt the signal strength of the transmission, the nearest with forward progress scheme [16] can reduce the packet collision where the packet is transmitted to the nearest neighbor of the sender which is closer to the destination.

### 4.3.6 Directional Flooding

In DREAM directional flooding [13], the sender $S$ of a packet with destination $D$ will forward the packet to all one-hop neighbors that lie "in the direction of $D$." In order to determine this direction, a node calculates the region that is likely to contain $D$, called the *expected region*. The expected region is a circle around the position of $D$ as it is known to $S$. The radius of the expected region is decided based on the time difference between the current time and the timestamp of the location information of $D$ and the maximum speed $v$ of $D$.

## 4.4 Reputation Systems

The goal of a reputation system is to detect whether the participating nodes in a MANET are behaving properly. Each node is assumed to be able to listen to the transmissions of its neighbors in promiscuous mode [17]. Reports are exchanged between nodes.

### 4.4.1 Watchdog and Pathrater

The watchdog and pathrater proposed in [17] are used to identify misbehaving nodes. When a node transmits a packet, the node's Watchdog confirms that the next node along the route also forwards this packet. The watchdog declares the neighboring node as misbehaving if it fails to forward certain number of packets within a specified time interval. This scheme is suitable for source routing protocols (e.g., Dynamic Source Routing [18]) to avoid message dropping attacks. The pathrater of a route selection scheme chooses a route by avoiding the misbehaving nodes.

### 4.4.2 Collaborative Reputation Mechanism (CORE)

Collaborative Reputation mechanism (CORE) [19] uses a reputation mechanism that differentiates between subjective reputation (observations), indirect reputation (positive reports by others), and functional reputation (task-specific behavior), which are weighted for a combined reputation value that is used to make decisions about cooperation or gradual isolation of a node. Reputation values are obtained by regarding nodes as requesters and providers, and comparing the expected result to the actually obtained result of a request.

### 4.4.3 CONFIDANT

The Cooperation Of Nodes: Fairness In Dynamic Ad hoc NeTworks (CONFIDANT) [20][21] protocol works as an extension to the reactive source-routing protocol to determine and isolate the selfish (and/or misbehaved) nodes. The *monitor* in each node observes the routing and forwarding behaviors of its neighbors. When suspicious events occur frequently more than the occurrence threshold value, the *reputation manager* updates the reputation value for the

corresponding neighbor. If the reputation value reaches the tolerance threshold value, the *path manager* deletes all routes containing this node from its path cache to avoid the suspected misbehaved nodes. At the same time, the *trust manager* sends ALARM messages to the source node of each path and to its friendly nodes.

## 4.5 Security Threats and Requirements for Position-based Routing

In this section, we first analyze the security threats of several geographic forwarding schemes and location services. We then describe the security requirements. Attackers can be classified into three primary categories: *malicious*, *compromised*, and *selfish* users (see Section 3.4).

### 4.5.1 Attack Models

Several attacks can be launched against the position-based routing protocols. These attacks include:

A1. *Message tampering attack*: Attackers can access the contents of messages and forward them with modified information. Since the forwarding decision is based on the destination's Location Information (LI) contained in messages, attackers can alter the LI in messages to disrupt the operation of unicast forwarding scheme. As shown in Figure 4.3(a), assume two paths exist between *B* and *A* via *C* (i.e., path BCEA and path BCFDEA). When node *C* receives a message *m* from *B*, it can modify the LI of *A* and forward modified message *m'* to other colluding node *D* via node *F*. When node *D* receives *m'*, it will return re-modified message *m"* to *C* again, and so on. This makes a routing loop where messages traverse nodes in a cycle without being relayed to the real destination *A*. Each node disseminates the Location Update (LU) message periodically to update the LI maintained either by specific Location Servers (LSs) [3][12]–[14]. When an attacker *C* receives an LU message of node *A* [see Figure 4.3(b)], it can modify the LI of *A* and forward this modified message LU' to its neighbors. Thus, an attacker *C* can cause other nodes to fail to find a

route to $A$ if they are more than one hop away from $A$.

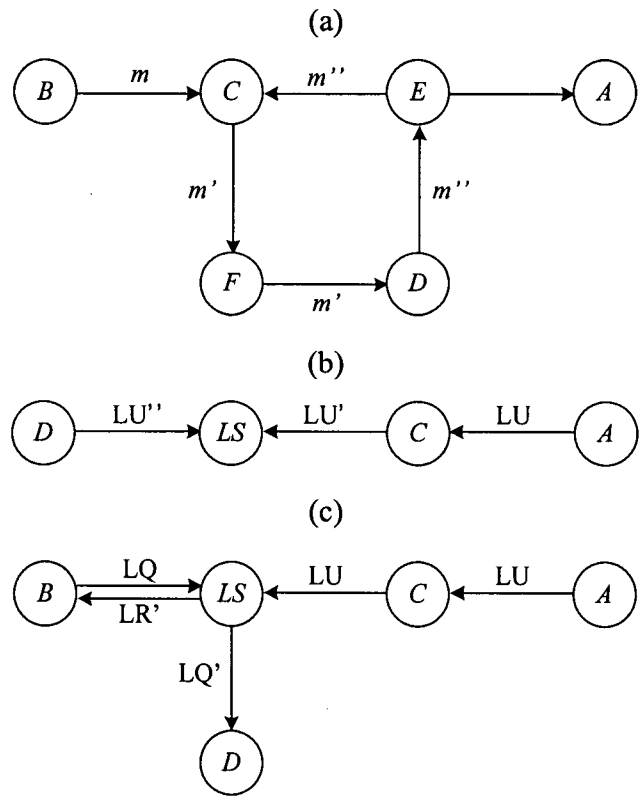



Figure 4.3 (a) Loop generation by changing location information; (b) message tampering against location update process; and (c) message tampering against location query process.

A2. *Message dropping attack*: A message dropping attack is feasible in all geographic forwarding schemes and location services. Both attackers and selfish nodes can drop some (or all) control or data messages either to disrupt the operation of position-based routing protocol or to save its resource from forwarding messages for others. To increase the power of *message dropping* attack, a compromised user may try to redirect traffic toward itself. However, unlike topology-based routing protocols, an attacker may not be able to increase the chance to join a path in position-based routing protocols. That is because the location discovery and data forwarding paths may be different. Moreover, the data forwarding path keeps changing due to mobility. However, compromised users can disrupt the operation of

location service by dropping some control messages [e.g., Location Query (LQ), Location Reply (LR), Location Error (LE) messages in GLS], which are transmitted in the form of a single unicast message.

A3. *Falsified message injection attack*: As shown in Figure 4.3(b), an attacker *D* can impersonate *A*, and generate a falsified message LU" with the latest timestamp. As a result, even a single attacker *D* can cause other nodes to fail to find a route to *A* if they are more than one hop away from *A*. As shown in Figure 4.3(c), to perform a location discovery to node *A*, node *B* sends an LQ message to *A*'s LS. If the LS is compromised, it can disrupt the location discovery process. By attaching the fake LI of *A*, the modified LQ' message may be forwarded to node *D*. As far as other nodes are concerned, the location service is functioning normally. Therefore, this attack cannot be detected by providing either *sender authentication* or *message integrity*. This attack is only feasible by compromised users.

A4. *Message replay attack*: Attackers can re-play (or re-transmit) eavesdropped messages again sometime later in a different place. This attack can consume either network resources (e.g., bandwidth) or node resources (e.g., memory, computation power). Although a wormhole attacker [2] can tunnel an LQ message directly to a destination node, it cannot always prevent other routes from being used for data communication. This is because the location discovery and data forwarding paths may not be the same in position-based routing protocols.

### 4.5.2 Security Requirements for Position-based Routing Protocols

Our secure position-based routing protocol prevents each of the threats mentioned in Section 4.5.1. To guarantee successful location service and data forwarding, a secure position-based routing protocol must provide the following security services:

1. *Source authentication*: The receiver should be able to confirm that the identity of the source is indeed who or what it claims to be.

2. *Neighbor authentication*: The receiver should be able to confirm that the identity of the sender (i.e., one hop previous node) is indeed who or what it claims to be.
3. *Location Information (LI) authentication*: In the presence of compromised users, the source node should trust the LI in the location table of LS only if the authenticity of LI is correctly verifiable.
4. *Message integrity*: The receiver should be able to verify that the content of messages has not been altered either maliciously or accidentally in transit.
5. *Access control*: The sender should be able to detect and prevent attackers from joining a path as a next hop node.

By providing *neighbor authentication*, malicious users cannot either drop or tamper messages because they cannot join a communication path. When source node provides the security services of both *source authentication* and *message integrity* to each message, intermediate nodes cannot modify the LI in received messages. Each node should be able to check the correctness of the LI of node $A$ in a received message, which is generated by one of $A$'s LSs. We call this security service a *location information authentication*. To detect and isolate attackers who cannot be prevented completely by using a cryptographic mechanism (e.g., *message dropping* attacker), the security service of *access control* is necessary. In the context of MANET routing, *confidentiality* is not a critical component in non-military scenarios. Some researchers argue that *non-repudiation* can be used for isolating misbehaving nodes in MANET routing protocols [22].

## 4.6 Network Environments

Our proposed secure protocols aim to protect the network layer from attackers. Attacks in other layers (e.g., physical, transport, and application) are beyond the scope of this chapter. Our proposed schemes work under several assumptions. These assumptions are stated as follows:

1. The network links are bi-directional. That is, if node $A$ is able to transmit to node $B$, then $B$ is

also able to transmit to $A$.

2. The wireless interface supports promiscuous mode operations. That is, each node can receive a copy of the message being transmitted by other nodes within its receiving range.
3. All nodes have tightly synchronized clocks with the maximum synchronization error of $\Delta$.
4. There exists a public key infrastructure in MANETs. Each mobile node stores the trusted Certification Authority (CA)'s public key.

We now explain the rationale behind these assumptions. The first assumption is common in practice. Many wireless medium access control protocols require bi-directional links to exchange several link-layer frames between a sender and receiver to avoid collisions. The second assumption is feasible since wireless interface cards nowadays support the promiscuous mode. Each node continuously monitors its neighbor's transmission in order to detect misbehaving nodes. For the third assumption, most of the position-based routing protocols require each node to have GPS to obtain its own LI. The accurate time synchronization can be maintained with embedded GPS. Some hardware clocks can be used in special applications to provide sufficiently accurate time synchronization (e.g. 183 ns) for several months [8]. Although the time synchronization signal itself may be subject to attack, we assume that each node can verify the correctness of its own position, and the neighbor's LI can be verified by using location verification techniques [23].

For the fourth assumption, the CA's public key is used to authenticate the public TESLA (and TIK) keys for hash chain of other nodes and to set up shared secret key between source and destination [24][25]. The methods of certificate-based public key distribution have been discussed in Section 3.5. The same assumption is used in some previous work [26][27]. Based on the public key infrastructure, any source and destination pair can set up the shared secret key before data transmission by using an authenticated Diffie-Hellman key exchange protocol [28]. This protocol was developed to defeat the *man-in-the-middle* attack [29] on the

Diffie-Hellman key agreement protocol. The immunity is achieved by allowing the two parties to authenticate themselves to each other by the use of digital signatures and public key certificates. There are also several offline methods to exchange the secret information. For example, a source node can exchange the secret key over different channels such as the telephone, e-mail, carrier service [29].

## 4.7 Secure Geographic Forwarding (SGF)

We distinguish two main forwarding strategies: *greedy forwarding* (i.e., unicast) [11] and *directional flooding* (i.e., broadcast) [13]. In this section, we assume that the source node has already obtained the position of the destination.

We use the following notations in this chapter:

1. $K_A^{TI}(j)$ [or $K_A^{TE}(j)$] denotes the TIK (or TESLA) key of node $A$ at the $j^{th}$ interval; $K_{AB}$ denotes the shared secret key between nodes $A$ and $B$; and $K_A$ denotes the private key of node $A$.
2. $MAC_K(M)$ denotes the MAC of message $M$ with a symmetric key $K$ using the Hashed MAC algorithm [30].
3. $Sign_K(M)$ denotes the digital signature of a message $M$ with the private key $K$ using the public key cryptography [31].

### 4.7.1 Secure Geographic Forwarding (SGF) with Unicast Messages

In this section, we propose the use of MAC computed over the non-mutable part (e.g., the LI of destination) of unicast messages with the pair-wise shared secret key between source and destination. Since intermediate nodes do not have the shared secret key with source node, they cannot verify the non-mutable part of messages. This allows a compromised user to be able to modify the non-mutable part of messages to disrupt the operation of position-based routing protocol. To prevent this attack, source node can use the digital signature over the non-mutable part with its own private key instead of MAC. However, implementing a mechanism to sign the

non-mutable parts of all data and control messages may introduce too much overhead. In our scheme, we propose the use of a reputation system (see Section 4.9) to detect and isolate *message tampering* and *dropping* attackers instead of using expensive digital signatures.

We propose to use the TIK protocol [2] with tight time synchronization to authenticate a previous forwarding node to prevent malicious users from joining a path and to avoid a *message replay* attack. Based on the third assumption stated in Section 4.6, each node can estimate the TIK key expiration interval *t_{disclosure}* (see Section 4.10.1). In addition, every node has its own one-way key hash chain.

Our proposed Secure Geographic Forwarding (SGF) mechanism works as follows. When a source node $S$ sends a message via its neighbor to a destination $D$, each intermediate node $i$ (i.e., sender) forwards the following message:

$$< MAC_{K_i^{TI}(j)}[M_i \,||\, N_S \,||\, MAC_{K_{SD}}(N_S)], MAC_{K_{SD}}(N_S), M_i, N_S, K_i^{TI}(j) > \qquad (4.1)$$

where $M_i$ represents the mutable parts of message from sender $i$, and $N_S$ represents the non-mutable part of message from source $S$. The notation $i$ is equal to $S$ when the sender is the source node itself. The sender $i$ discloses the key $K_i^{TI}(j)$ at the end of the same message. Figure 4.4 shows the timelines of sending and receiving a SGF message between two neighbors. Time $t_i$ indicates the time when sender $i$ starts transmitting the message, and time $t_i$ + *t_{disclosure}* is the disclosure time for key $K_i^{TI}(j)$.

Because of time synchronization, when the neighbor receives the message portion $MAC_{K_i^{TI}(j)}[M_i \,||\, N_S \,||\, MAC_{K_{SD}}(N_S)]$, it can verify that the sender $i$ has not started sending the corresponding key $K_i^{TI}(j)$ if the following condition is satisfied:

$$t\_\{disclosure\} \le \tau - \Delta + Q/r \qquad (4.2)$$

where $\tau$ is the propagation delay; $Q$ is the size of the message excluding $K_i^{TI}(j)$; and $r$ is the transmission rate. As the receiver knows the expiration time for each key and the sender $i$ only

discloses the key after it expires, the attackers cannot guess the value of $K_i^{TI}(j)$. Therefore, if the message authentication verifies correctly once the receiver later receives the authentic key $K_i^{TI}(j)$, the message must have originated from the claimed sender. Since only the sender knows the key $K_i^{TI}(j)$, at the time when the receiver received the message, other nodes cannot forge a new message with the correct MAC. Finally, when destination $D$ receives this message, it can verify the authenticity of the message by comparing the received $MAC_{K_{SD}}(N_S)$ to the MAC value that is computed over the received message $N_S$ with the secret key $K_{SD}$ it shares with the source node $S$. Each node re-establishes its authentic TIK key every $t_h$-second with its neighbors by piggybacking on a HELLO message of SGLS.

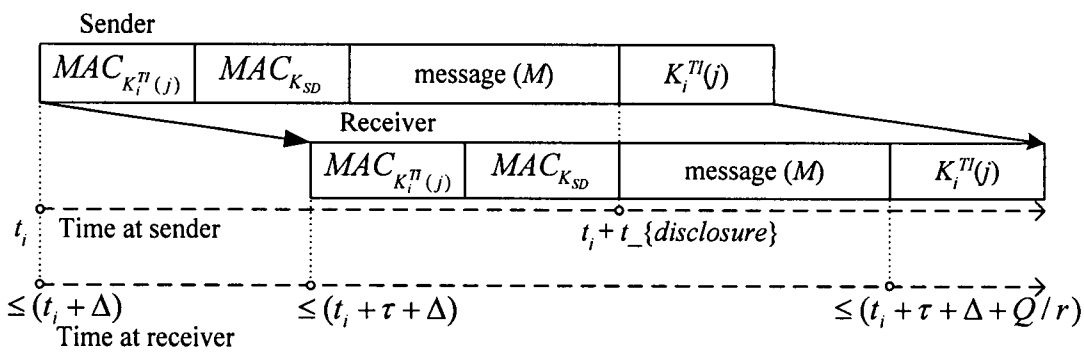



Figure 4.4 Secure Geographic Forwarding (SGF) of a unicast message.

Note that although there are several forwarding strategies [15][16], they all forward a given message to only one optimal neighboring node based on its optimization criterion. Therefore, our proposed SGF can be applied to any of these forwarding schemes without any modification.

### 4.7.2 Secure Geographic Forwarding (SGF) with Directional Flooding

There are several directional flooding mechanisms [13] based on the determination of expected region. Although messages are flooded to multiple neighbors, each message is still a "*unicast*"

message with the same destination address. Therefore, we can use the same scheme proposed in Section 4.7.1 for each unicast message.

### 4.7.3 Discussion

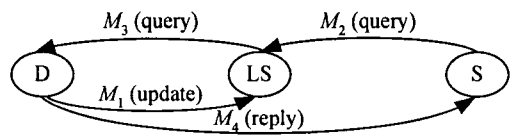
When the node's clocks cannot be tightly synchronized, it is impossible to use a TIK protocol for *neighbor authentication*. In that case, each intermediate node should sign the whole message together with a timestamp by using a digital signature scheme [31]. Since a limited clock skew can be maintained with any off-the-shelf GPS, we can avoid *message replay* attacks by malicious users. Another possible solution can be provided by using MAC with a shared secret key between all nodes along the path including source and destination. The computation of MAC is very efficient and fast, even affordable for low power devices. However, since both the location discovery and data forwarding paths are different in position-based routing protocol, the establishment of the secret key between all nodes is a non-trivial problem. Moreover, any intermediate node with the same secret key can impersonate the source node.

Since neither honest nodes nor central authority can distinguish messages injected by compromised users from messages generated by honest nodes, it is difficult to avoid *message injection* attack. In a greedy forwarding scheme, since each message cannot flood the whole network, we do not explicitly protect against this message injection. On the other hand, in directional flooding, since a falsified message from compromised users can flood part of the network, injecting many messages can result in a DoS attack. To reduce the power of this attack, the rate of generating messages should be limited. Thus, neighboring nodes can filter out excessive messages immediately.

## 4.8 Secure Grid Location Service (SGLS)

In this section, we describe our proposed SGLS protocol based on SGF. SGLS provides several security mechanisms to the original GLS. Figure 4.5 summarizes the operation of SGLS in

combination with SGF. Our proposed SGF concept developed in Section 4.7.1 can generally be applied to any unicast message such as Location Query (LQ), Location Reply (LR) and Location Error (LE) messages of GLS.




$M_1 =< h_1, \text{UPDATE}, Sign_{K_D}(N_{UPDATE}), K_D^{TI} >$ where $h_1 = MAC_{K_D^{TI}}[\text{UPDATE}, Sign_{K_D}(N_{UPDATE})]$

$M_2 =< h_2, \text{QUERY}, MAC_{K_{SD}}(N_{QUERY}), K_S^{TI} >$ where $h_2 = MAC_{K_S^{TI}}[\text{QUERY}, MAC_{K_{SD}}(N_{QUERY})]$

$M_3 =< h_3, \text{QUERY}, MAC_{K_{SD}}(N_{QUERY}), Sign_{K_D}(N_{UPDATE}), K_{LS}^{TI} >$ where $h_3 = MAC_{K_{LS}^{TI}}[\text{QUERY}, MAC_{K_{SD}}(N_{QUERY})]$

$M_4 =< h_4, \text{REPLY}, MAC_{K_{SD}}(N_{REPLY}), K_D^{TI} >$ where $h_4 = MAC_{K_D^{TI}}[\text{REPLY}, MAC_{K_{SD}}(N_{REPLY})]$

Figure 4.5 Location update and query in SGLS where UPDATE, QUERY, and REPLY denote the original GLS's location update, location query, and location reply message, respectively; $N_X$ represents the non-mutable fields of message $X$.

### 4.8.1 Secure Location Update and Query between Destination and Location Server

Unlike other messages, the LU message has no assigned destination address field in it. Thus, it is impossible to provide a *source authentication* with a symmetric secret key. Moreover, as mentioned in Section 4.5.1, when a source node sends an LQ message to one of $D$'s LSs, LS can disrupt the location discovery process by attaching the fake LI of $D$ to the LQ message. To protect the LU message, a destination node $D$ attaches the digital signature computed over the non-mutable part (e.g., LI of a destination) of an LU message. At the same time, the TIK protocol is used for *neighbor authentication* as described in Section 4.7.1.

After receiving a valid LU message from $D$, LS stores the digital signature of $D$ in its location table. When the LQ message generated by $S$ toward $D$ arrives at this LS, LS can prove that it has the valid LI of $D$ by attaching $D$'s digital signature from its location table to the LQ message. Thus, LS can provide an *LI authentication* to all intermediate nodes along the path to $D$

(see Figure 4.5).

A destination node $D$ sends the LU messages at a rate proportional to its speed $v$ and the distance $d$ to the square of LS. To guarantee the freshness of LI, $D$ attaches the lifetime $2^{i-2} \cdot d/v$ to its LU message toward order-$i$ LS where $i \geq 2$ in GLS. In our proposed scheme, the lifetime value is digitally signed together with the LI of $D$ to avoid unexpected changes. If any intermediate node overhears either an incorrect digital signature or the expired lifetime of the LI, it will invoke the reputation system (see Section 4.9) to indicate that a previous forwarding node intentionally has changed the LI of $D$. Thus, this message will be dropped accordingly.

### 4.8.2 Secure Location Query from Source to Location Server

A location query can fail when an intermediate node is either compromised or selfish. For example, the LS of a destination node $D$ can drop an LQ message without sending an LE message to a source node $S$. Although we can force the LS to generate an LE message by using a Local Reputation System (LRS), an LQ dropping may result in multiple location discoveries. Furthermore, the second LQ message may be forwarded to the same LS again because a destination node $D$ always recruits a node with the closest ID as its LS. Thus, a source node $S$ cannot find the location of $D$ until a good node replaces that compromised LS in that square. To solve this problem, we propose to include the LI of the broken link in an LE message. Note that compromised node cannot cheat its own LI due to its neighbor's LRS. When $S$ receives this message, it can avoid the suspicious node by indicating the level of hierarchy and the location of square to forward next LQ message. For example, when $S$ receives the LE message from LS2 located in the order-2 square on the left bottom in Figure 4.2, $S$ will search the order-2 square on the right bottom first in its next location discovery process.

### 4.8.3 Secure Exchange of HELLO Messages

In GLS, each node maintains a table of its immediate neighbors as well as each neighbor's

neighbors. The one-hop neighbor's LI can be verified by using a location verification technique [23], and the TIK protocol can be used for *neighbor authentication*. However, the LI about a neighbor's neighbors cannot be verified by using these techniques since they are out of the transmission range of the verifier.

We propose to use the TESLA [4] broadcast authentication method to verify the LI of two-hop neighboring nodes. For example, a node $A$ includes two additional fields in a HELLO message: $< MAC_{K_A^{TE}(j)}(LI_A), K_A^{TE}(j-1) >$ where $LI_A$ is the location information of node $A$; and $K_A^{TE}(j-1)$ is the TESLA key of $A$ at the $(j-1)^{th}$ time interval. Since a HELLO message is broadcast periodically with interval $t_h$, the TESLA key disclosure interval can be set to the value of $t_h$.

When a two-hop neighbor node $C$ receives a HELLO message, it checks the validity of the LI of $A$ by determining that $K_A^{TE}(j)$ has not yet been disclosed. Node $A$ waits until it is able to disclose $K_A^{TE}(j)$ from the time interval schedule; it then appends $K_A^{TE}(j)$ to the next HELLO message. When node $C$ receives a new HELLO message, it can verify the previous LI from $A$. If this verification process fails, the LRS is called upon to report the fact that neighbor $B$ intentionally changes LI of its neighbor $A$. One limitation of this scheme is that two-hop neighbors' LI can only be verified correctly after time $t_h$ at a maximum.

### 4.8.4 Discussion on Other Location Services

Although a number of different location services were developed recently [10], most of location services are the variations and combinations of the following four schemes: Grid Location Service (GLS) [3], Quorum based location service [14], Virtual Home Region (VHR) [12], and DREAM location service [13]. Since VHR has exactly the same security problems with GLS, we investigate only quorum-based location service and DREAM in the following sub-section.

#### 4.8.4.1 Quorum based Location Service

Since the virtual backbone nodes maintain interconnection among themselves by using any topology-based routing methods, quorum-based location service has security threats against the corresponding routing protocol. Moreover, since backbone nodes have a shared responsibility to maintain the LI of all other nodes, attackers may want to join a backbone with an intention to increase the chance of attacking. Depending on the rules of choosing backbone nodes, an attacker can send advertisement messages with fake information of neighboring nodes via flooding during the initial setup of the virtual backbone. To countermeasure this attack, we can use the TESLA based scheme proposed in Section 4.8.3. Thus, receiving node can verify whether the advertisement message includes any invalid information of neighboring nodes. Since all attacks mentioned in Section 4.5.1 are feasible in this scheme, security mechanisms proposed in Sections 4.7 and 4.8 can be applied with a minor modification.

#### 4.8.4.2 DREAM Location Service

Since each node floods the LU message to the whole network, this scheme is less resilient against DoS attacks such as *broadcast message injection* attack to exhaust the network's computation resources. A broadcast authentication protocol [4] is required to enable the receivers to verify that the broadcast messages they received were actually sent by the claimed sender as shown in Section 4.2. Unlike other location services, there is no specific LS in DREAM. Therefore, there is no threat of cheating the LI of destination only if the *source authentication* can be provided.

## 4.9 Local Reputation System (LRS)

As we mentioned in Section 4.5.1, compromised users can disrupt the operation of location service by dropping some control messages, which are transmitted in the form of single unicast. Moreover, if there is no punishment for misbehaviors, attackers may be rewarded and encouraged to attack again later. In this section, we propose the reputation system with an aim to

detect and isolate attackers. We extend the reputation system (i.e., CONFIDANT) originally proposed by Buchegger and Le Boudec [20][21] and modify it to work specifically for position-based routing protocols. We call our extended version as the Local Reputation System (LRS).

Both the original CONFIDENT reputation system and our proposed LRS use the same set of mathematical equations for reputation report update. However, the CONFIDENT reputation system assumed the use of the source routing protocol. Various ALARM messages are sent to the source node when anomaly was detected. On the other hand, in our proposed LRS, we assume the use of position-based routing protocols. Each node periodically sends the reputation information report to its neighbors by using the HELLO message. In LRS, each node only needs to manage the reputation information of its local neighbors.

LRS consists of the following three components: *the monitor*, *the reputation manager*, and *the trust manager*. All these components are present in each node. Different modules in each component are shown in Figure 4.6.

### 4.9.1 First-Hand Reputation Rating

Node $i$ maintains a record of the *first hand observation* about node $j$ in the form of $F_{ij}^{l} = (\alpha, \beta)$ = (*# of good behaviors*, *# of bad behaviors*) for the $l^{th}$ reputation interval, and is initially set to (1, 1).

For example, if the observation is classified as misbehavior, the value of $\beta$ is increased by one. The first-hand reputation rating is represented in the form of $FR_{ij}^{l} = \alpha/(\alpha+\beta)$ [20]. When the reporting timer expires, the first-hand reputation information $FR_{ij}^{l}$ about node $j$ from node $i$ is updated as follows: $FR_{ij}^{l} = \upsilon \cdot FR_{ij}^{l-1} + (1-\upsilon)\cdot FR_{ij}^{l}$ where $\upsilon$ is a weighted value. During inactivity periods, the value is updated periodically as follows: $FR_{ij}^{l} = \upsilon \cdot FR_{ij}^{l-1} + (1-\upsilon)\cdot FR_{\_initial}$ where $FR_{\_initial}$ is 0.5.

### 4.9.2 Reputation Reporting and Second-Hand Reputation Rating

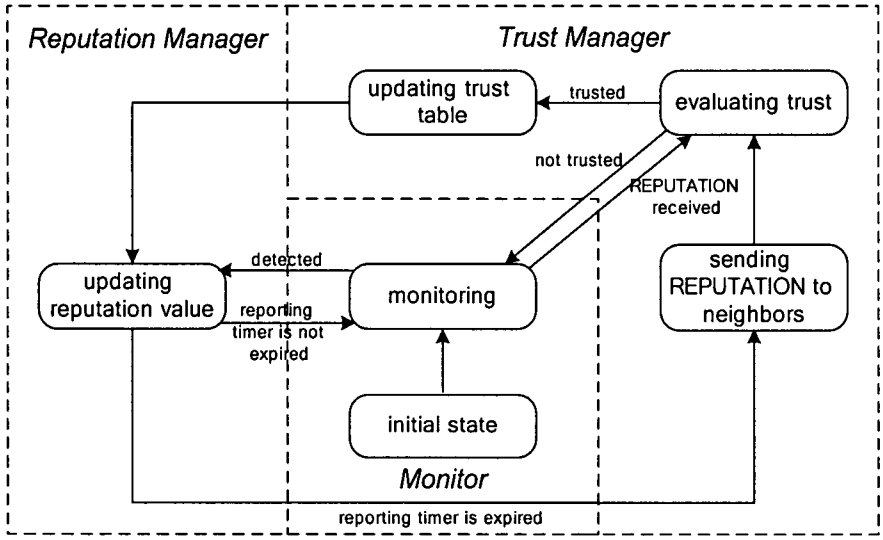



Figure 4.6 Structure of Local Reputation System (LRS).

A node's reputation information is sent periodically to its neighbors by piggybacking on a HELLO message when the $l^{th}$ reporting timer expires. Assume node $i$ receives the reported *second-hand reputation information* $FR_{kj}^{l}$ about node $j$ from node $k$, node $i$ updates the reputation rating $R_{ij}^{l}$ as follows: $R_{ij}^{l} = (1-\omega)\cdot FR_{ij}^{l} + \omega\cdot FR_{kj}^{l}$ where $\omega$ is a small positive real number. This process is performed for all $j$ being reported. Based on this reputation rating, node $i$ classified node $j$ as a good node if $R_{ij}^{l} \geq \gamma$; or as a bad node if $R_{ij}^{l} < \gamma$ where $\gamma$ is a predefined threshold value. To avoid blackmail attack, our reputation system can also take into account the trust rating of each node [20]. For simplicity, we do not consider the false report of reputation in this chapter.

### 4.9.3 Countermeasures for Message Tampering and Dropping

There are two attacks that can be partially defended by LRS: *message tampering* and *dropping attack*. Figure 4.7 illustrates how the monitor module works. Suppose there is a path from node $S$ to $D$ through intermediate nodes $A$, $B$, and $C$. Node $A$ cannot transmit directly to $C$, but it can listen in on $B$'s traffic. Thus, when $A$ transmits a message for $B$ to forward to $C$, $A$ can find out whether $B$ relays the correct message or not.

Table 4.1 Pseudo code for monitoring module.

```
A node has sent or overheard a message:

Case I: Unicast Message
1.  if (dst addr == my addr)
        return;
2.  if (src addr == my addr)
        buffer a message; return;
3.  if (next hop addr == one of my neighbors' addrs && next hop addr != dst addr)
        if (overheard the message with the same ID before)
            if (contents are correct)
                call reputation manager with positive feedback; remove this entity; return;
            else
                call reputation manager with negative feedback; remove this entity; return;
        else
            buffer a message and return;
4.  if (timer expires for any message in buffer)
        call reputation manager with negative feedback; remove this entity; return;

Case II: HELLO Message
1.  if (src addr == my addr)
        buffer a LI of mine; return;
2.  if (LI of mine is correct)
        call reputation manager with positive feedback; return;
    else
        call reputation manager with negative feedback; return;
```

In Figure 4.8, the monitor module of node *A* can also overhear a message generated from *B* where the next hop address field is matched with one of its neighbors' addresses, and is not the final destination. The monitor module maintains a buffer of recently either sent or overheard messages and compares each overheard message with the message in its buffer to see if there is a match. When a message has remained in the buffer for longer than a timeout interval or the content of message has changed maliciously, the reputation manager is called. The

reputation manager then decreases the reputation value for the node responsible for forwarding the message. When the reporting timer expires, the node sends its first-hand reputation information in its HELLO message to warn its neighbors of attackers. In this way, our LRS can detect both message tampering and dropping attacks. Table 4.1 describes the pseudo code for the monitoring system of LRS.

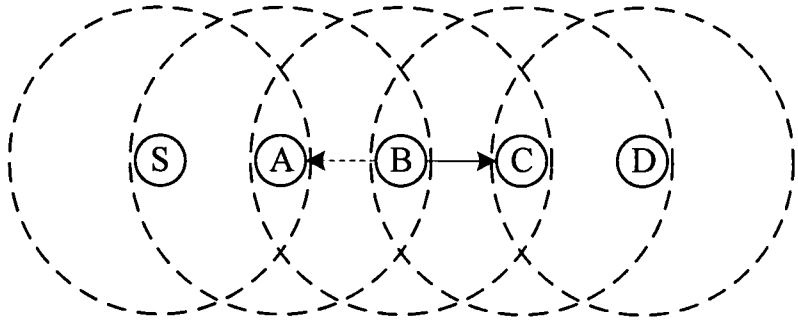



Figure 4.7 Promiscuous mode for monitoring. Dashed circle represents the transmission range of each mobile node. Dashed line indicates that node *A* can overhear *B*'s transmission to node *C*.

### 4.9.4 Limitations of Reputation System

In general, it may be difficult to distinguish misbehaving from transmission failures and other kind of failures [17] in wireless channels. A reputation system only provides probabilistic guarantees of the detection of misbehaving nodes. Although the reputation system with both positive and negative feedbacks can force misbehaving nodes to behave correctly up to the certain threshold level, it is impossible to avoid *blackmail attack* completely. For example, an attacker can first participate in the routing and data forwarding operations properly in order to increase its reputation and trust ratings to exceed certain threshold levels. After that, it can send the falsified reputation messages to the network. One feasible solution is the use of Tamper Resistant Module (TRM) to protect the routing modules. The use of TRM can be justified due to the seriousness of blackmail attacks in hostile environments such as military battlefields.

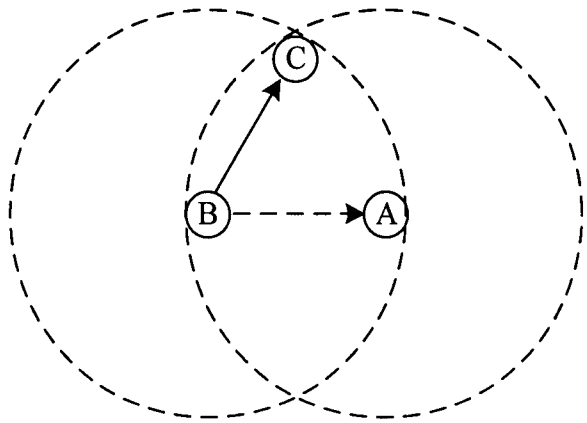



Figure 4.8 Node *A* can overhear and buffer the transmission from node *B* to *C* where node *C* is not the final destination. If this packet has remained in the buffer for longer than a certain timeout, the reputation manager will be called.

## 4.10 Performance Evaluation

To evaluate the suitability of SGLS for use in MANET, we analyze the computational complexity of its TIK operation. We also conduct simulation experiments to evaluate the performance of our proposed SGLS without attackers and LRS with attackers.

### 4.10.1 Performance of TIK in Secure GLS

As mentioned in Section 4.7.1, SGLS uses the TIK-based geographic forwarding mechanism for *neighbor authentication* of received messages. However, in a contention-based medium access control protocol, a potential problem of using TIK is that a sending node cannot predict the precise time of message transmission. For instance, in IEEE 802.11b medium access control protocol, the sending node cannot know the exact time of message transmission until one slot time (20 μsec) [9] before the transmission. Therefore, the generation time for MAC must be smaller than this slot time, and TIK mechanism must be implemented in the medium access control protocol.

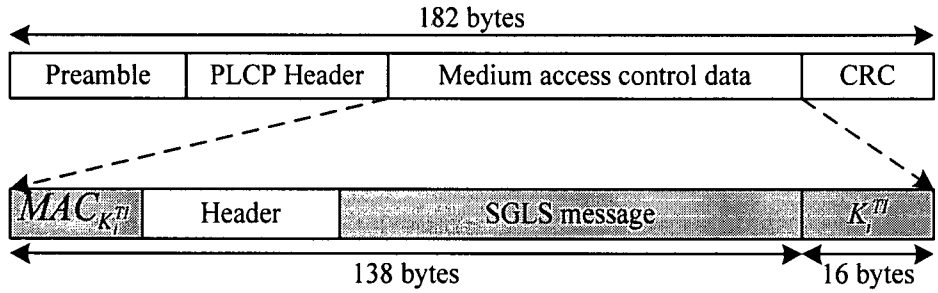



Figure 4.9 Minimum size of IEEE 802.11b frame format in SGLS where PLCP stands for Physical Layer Convergence Protocol.

Since the sender discloses the key in the same message that carries the corresponding MAC, a key disclosure interval *t_{disclosure}* must be chosen to locate somewhere during the message's transmission (as shown in Figure 4.4). Therefore, this key disclosed interval must be decided according to both the minimum payload length of a frame and the bit rate of transmission. Assuming that both the propagation delay $\tau$ and the maximum time sychronization error $\Delta$ are negligible, a key expiration time interval *t_{disclosure}* can be determined with the ratio of the size of message to transmission rate as shown in equation (4.2). The current wireless LAN products, such as commonly used IEEE 802.11b cards, provide a transmission data rate of 11 Mbps. With a 138-byte minimum message size [i.e., link layer header (38 bytes), IP header (20 bytes), UDP header (8 bytes), and SGLS message (72 bytes)] as shown in Figure 4.9, the key expiration time interval *t_{disclosure}* must be around 100 μsec. To verify the received unexposed key $K_i$ with a known previous key $K_j$ where $j < i$, by $K_j = h^{i-j}[K_i]$, a receiving node needs to evaluate $i - j$ hash functions.

The number of hash function evaluations per second can vary depending on its implementation. For example, on the Pentium III 800 MHz, the sender can compute approximately $10^6$ MD5 hash function [7] evaluations per second (i.e., the software-based implementation) [32] The modified version of MD5 hash code [2]. evaluates the MD5 hash function at the rate of $1.3\times10^6$ times per second. Moreover, the speed of MD5 hash function

evaluation can be accelerated with the help of simple stand-alone hardware [33]. Assuming that the wireless link is error-free, and the channel is never idle [i.e., a data frame starts as soon as DCF Inter Frame Space (DIFS) after channel becomes free], we consider two nodes connected by an IEEE 802.11b wireless link at a data rate of 11 Mbps. The maximum number of transmissions can be achieved when the following cycle of transmissions is repeated without backoffs: RTS (44 bytes), SIFS (Short Inter Frame Space 10 μsec), CTS (38 bytes), SIFS, FRAME (182 bytes as shown in Figure 4.9), SIFS, ACK (38 bytes), and DIFS (50 μsec). Note that each frame (i.e., RTS, CTS, FRAME, and ACK) contains the PLCP overhead of 24 bytes. In addition, the PLCP overhead and control frames are always transmitted at 1 Mbps. Therefore, with the default parameters of IEEE 802.11b DCF protocol [9], the total transmission time for each cycle is 1346.9 μsec. Based on this calculation, the maximum number of message transmissions per second is about 742. In other words, a node can verify MAC in a message at most every 1346.9 μsec to keep up with link-speed 11 Mbps. When the key expiration time interval is 100 μsec, TIK can authenticate messages at link-speed 11 Mbps using only 10,000 hashes per second. This is less than 5 % load on CPU time even on the Compaq iPaq 3870 PocketPC, which is capable of performing 222,000 symmetric cryptographic operations per second [2].

### 4.10.2 Simulation Environment

We consider a network topology with 100 nodes randomly placed over a 1000×1000 ($m^2$) flat-grid. The size of an order-1 grid is 250×250 ($m^2$). We assume that 50 of these nodes are constant bit-rate data sources, each sending fixed size 128-byte messages at 4 messages per second for 200 seconds. Each simulation run takes 600 simulated seconds. The characteristics of each mobile node's radio interface approximate the Lucent WaveLAN, operating as a shared-medium radio with a nominal bit rate of 2 Mb/s and a nominal radio range of 250 $m$. For the medium access control layer, the IEEE 802.11 DCF is used. The propagation model combines both a free

space and a two-ray ground reflection models. Table 4.2 provides other simulation parameters. For simplicity, we assume that false reports of reputation do not occur in our simulations. A random waypoint model is used for the mobility model. Each node moves in a straight line towards the destination at a speed that is uniformly distributed from 0 to 10 m/s. For fair comparisons, identical mobility and traffic scenarios are applied to all protocols. Results are averaged over 11 simulation runs; the error bars represent the 95 % confidence intervals about the means in Figures 4.10–4.24.

Table 4.2 Simulation parameters.

| SGLS and LRS parameters | |
|---|---|
| HELLO message interval $t_h$ | 2 seconds |
| TIK key re-establishment interval | 2 seconds |
| TESLA key disclosure interval | 2 seconds |
| Reputation reporting interval | 10 seconds |
| First hand reputation weight value $\upsilon$ | 0.9 |
| Second hand reputation weight value $\omega$ | 0 or 0.1 |
| Threshold $\gamma$ | 0.5 |

To evaluate our proposed LRS as presented in Section 4.9, we modify the ns-2 grid package [34] and implementing both LRS and blackhole attackers. In the following results, LRS-S refers to LRS using both first and second-hand reputation information (i.e., $\omega = 0.1$), and LRS-F refers to LRS using only first-hand reputation information (i.e., $\omega = 0$). We compare both LRS-S and LRS-F with the original GLS. The performance metrics for evaluations are *message delivery fraction*, *average end-to-end delay of transferred data messages*, and *routing overhead*. These performance metrics are described in Section 3.9.

### 4.10.3 LRS with Data Message Dropping Attackers

Figures 4.10–4.12 show the simulation results with varying number of blackhole attackers who

drop data messages in the network. The pause time is equal to zero in this scenario. Figure 4.10 shows the message delivery ratio as a function of the number of blackhole attackers. Both LRS-S and LRS-F yield a higher message delivery ratio than GLS as the number of blackhole attackers increases. This shows that our proposed LRS can effectively detect and isolate blackhole attackers. As it uses also the second-hand reputation information, LRS-S works slightly better than LRS-F with faster detection. One interesting point here is that LRS has a lower delivery ratio than GLS when the number of blackhole attackers is zero. This is because the LRS cannot distinguish between malicious dropping and other droppings as mentioned in Section 4.9.4, some data packets may be dropped without being forwarded to a suspicious node.

Figure 4.11 shows that GLS incurs a lower routing control overhead than LRS. This is due to the fact that LRS can detect the blackholes and re-initiate the location query (or detour) to avoid these nodes. These additional location discoveries increase the routing control overhead.

Figure 4.12 indicates that GLS has a lower average end-to-end delay when compared with LRS. Since LRS incurs more routing control messages, the average end-to-end delay for data messages increases. Note that the average end-to-end delay of GLS decreases as the number of blackhole attackers increases. Since blackhole attackers drop data messages at the intermediate nodes and the dropped messages are not counted in the end-to-end delay calculation, the average end-to-end delay is decreased.

Figures 4.13–4.15 show the performance comparison with varying pause time (i.e., mobility), while keeping the number of blackhole attackers at 15 out of 100. Figure 4.13 shows that the delivery ratio does not increase or decrease remarkably in all protocols. This is due to the fact that our simulation network is not very congested. Therefore, the increase of control overhead due to high mobility does not affect the delivery fraction.

Figure 4.14 shows that the routing overhead of all protocols decreases as pause time increases (i.e., mobility decreases). That is because each node updates its closest location servers

every time it moves a particular threshold distance $d$ (100 $m$ in this chapter) since sending the last update. This indicates that a node sends out updates at a rate proportional to its mobility.

Figure 4.15 shows the average end-to-end delay for all three protocols increases as mobility decreases. We can find the reason from the fact that on average a longer path is obtained as mobility decreases in a MANET [27]. In our experiment, the average number of hops increases from 2.4 to 3.4 as pause time increases from 0 to 600 seconds. Note that in a highly congested network, the end-to-end delay may instead decrease as mobility decreases.

### 4.10.4 LRS with Data and Control Message Dropping

In MANETs employing topology-based ad hoc routing protocols, control packet dropping attacks may not be able to join (or attack) the communication session as an intermediate node. However, since location query and location reply pass through different paths, control packet dropping attacker can still join the communication session as an intermediate node in position-based routing. In this set of simulations, blackhole attackers can drop not only data packets but also control (i.e., location query) packets to disrupt a routing protocol.

Figures 4.16–4.21 show the simulation results with varying number of blackhole attackers, who drop both data and control packets. In Figures 4.16–4.18, the simulations employ a zero pause time. In Figures 4.19–4.21, the number of blackhole attackers is fixed at 15. Figure 4.16 shows the packet delivery ratio as a function of the number of blackhole attackers. LRS yields a higher packet delivery ratio than GLS as the number of blackhole attackers increases. This shows that our proposed reputation system can still isolate blackhole attackers even if they drop location query and reply packets. Due to the use of second-hand reputation information, LRS-S works better than LRS-F. One interesting point here is that the performance difference in delivery ratio between LRS-S and LRS-F is remarkable. This indicates that the inaccurate and slow detection of reputation information in LRS-F reduces the delivery ratio when attackers drop control packets.

Figure 4.17 shows that both GLS and LRS-F incur a much lower routing control overhead than LRS-S. That is due to the detection of blackholes and the restart of location discovery. This result shows that first-hand reputation information is not enough when blackhole attackers drop both control and data packets.

Results in Figure 4.18 indicate that both GLS and LRS-F have a lower average end-to-end delay when compared with LRS-S. This is due to the larger number of routing control packets incurred by LRS-S to overcome blackhole attacks.

Figure 4.19 shows that the delivery ratio decreases quickly in all three protocols as pause time increases. This indicates that our reputation system cannot work efficiently when attackers drop both data and control packets as mobility decreases. This is because of the limitation of selecting and querying LSs in GLS. Since a small subset of deterministic nodes work as LSs, some nodes cannot find specific destination location information if the LS is malicious. As mobility decreases, the LS is changed less frequently, thus making the situation worse.

Figure 4.20 shows that the routing overhead of all protocols decreases as pause time increases (i.e., mobility decreases). That is because each node updates its closest LSs every time it moves a particular threshold distance after sending the last update.

Figure 4.21 shows the average end-to-end delay for all three protocols decreases as mobility decreases. Although on average longer paths result as mobility decreases in a MANET, the average end-to-end delay of all three protocols decreases. Since blackhole attackers drop control packets at the intermediate nodes, frequently the path to the destination cannot be found, thus reducing the average end-to-end delay.

### 4.10.5 SGLS without Attackers

In this sub-section, to evaluate the SGLS without attackers, we implement SGLS, which includes the TIK, TESLA, digital signature, and MAC without LRS. By comparing SGLS with the

original GLS, we can examine the performance impact of adding security overhead, independent of the effect of attackers. Figures 4.22–4.24 show the simulation results without attackers. The TIK overhead (32 bytes; one MAC and one key) is introduced in each IEEE 802.11 data frame. The additional overhead of MAC (16 bytes), TESLA key (16 bytes), and two authentic keys for TIK (32 bytes; one for current key chain and the other for next key chain) is considered for HELLO message. The end-to-end MAC (16 bytes) is added to all unicast messages except the LU message (digital signature of 40 bytes using the Elliptic Curve Cryptography [35]).

Figure 4.22 shows that the message delivery ratio between SGLS and GLS. Adding security overhead in SGLS reduces the message delivery ratio by just 1 % on average. This suggests that SGLS is still effective (over 90 %) in discovering and maintaining routes for delivery of data messages even in relatively high mobility scenarios. It also implies that the traffic load of our simulation is low enough to tolerate the security overhead for SGLS.

Figure 4.23 shows that SGLS's routing overhead is much higher than that of GLS in terms of bytes. That is due to the increase in size of routing control messages with digital signature and MACs in SGLS. As mobility increases, the amount of control overhead of SGLS increases slightly. This result suggests that the usefulness of the secure routing protocol is closely related to the level of congestion and the amount of introduced overhead. Due to efficient cryptographic mechanisms, SGLS can still maintain high performance in our simulation environments.

Figure 4.24 shows that the average end-to-end delay for SGLS is slightly higher than that of GLS. Intuitively, SGLS may have a higher average delay for the location discovery than GLS. However, the number of location discoveries performed is a small fraction when compared with the number of data messages delivered. Therefore, the effect of the location acquisition latency on the average end-to-end delay of data messages is not significant.

## 4.11 Summary

In this chapter, we have proposed SGLS, which is a security enhancement to the original GLS protocol. The security mechanisms added to GLS include TIK, TESLA, MAC, digital signature, and a reputation system. SGLS has the capability of preventing *message tampering*, *dropping*, *falsified injection*, and *replay* attacks by either malicious or compromised users. To the best of our knowledge, this may be the first approach to address security issues for position-based routing protocols. Simulation results showed that in the presence of message dropping attacks, the proposed Local Reputation System (LRS) maintains a high message delivery ratio at the expense of a higher average end-to-end delay and routing overhead in general. We have also investigated the computational complexity of SGLS through analysis and simulations.

For the implementation of position-based routing protocols, each mobile device needs to have the GPS capability. If GLS is used for location service, the information such as the origin and size of the grid need to be known by the mobile devices a priori. When the proposed security enhancements need to be included, further changes including those listed in Section 4.6 need to be taken into account.

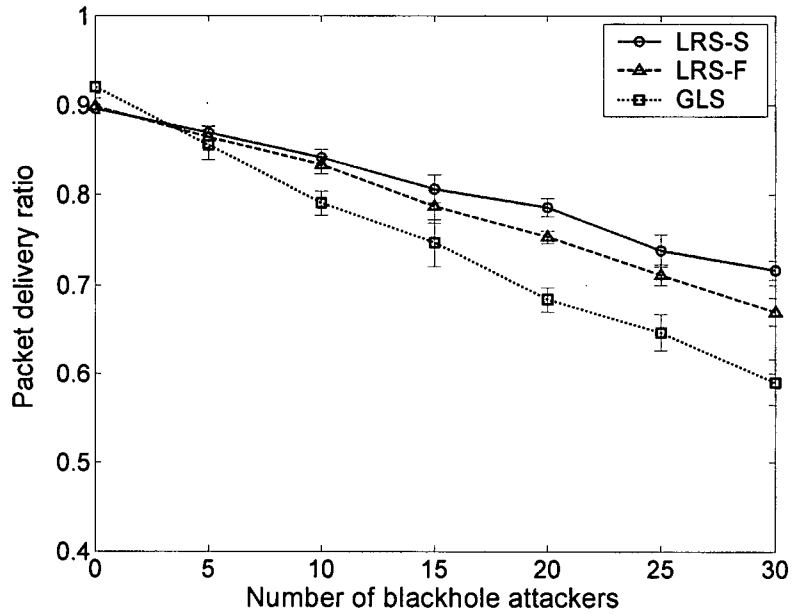



Figure 4.10 Packet delivery ratio between LRSs and GLS with varying number of data blackhole attackers (pause time = 0 sec).

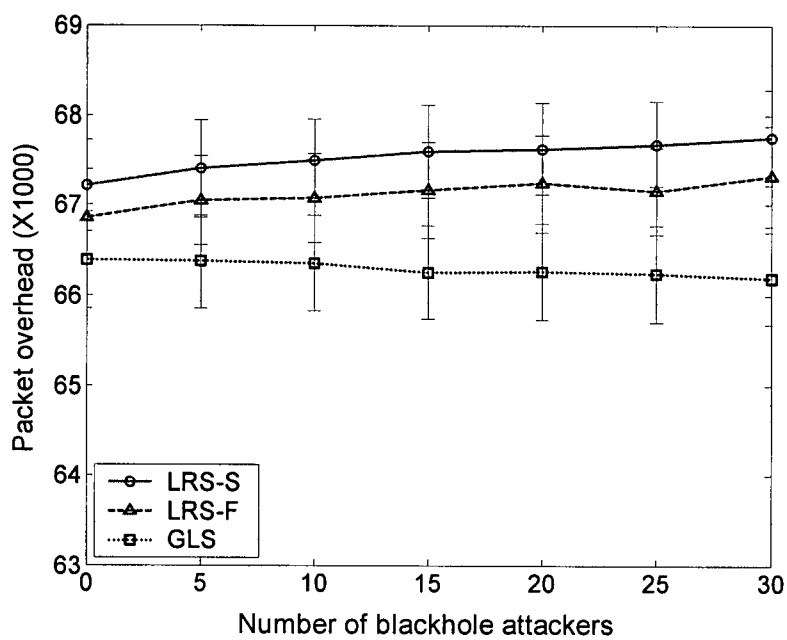



Figure 4.11 Routing packet overhead between LRSs and GLS with varying number of data blackhole attackers (pause time = 0 sec).

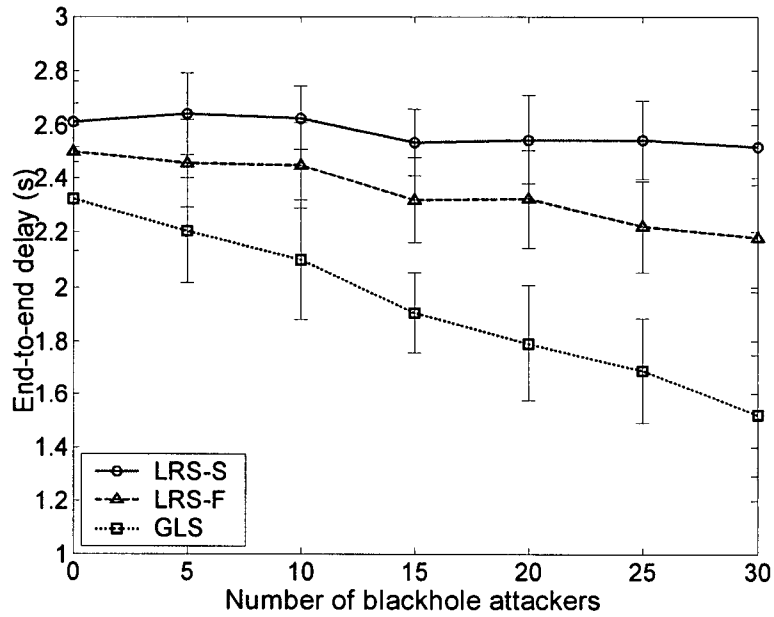



Figure 4.12 Average end-to-end delay between LRSs and GLS with varying number of data blackhole attackers (pause time = 0 sec).

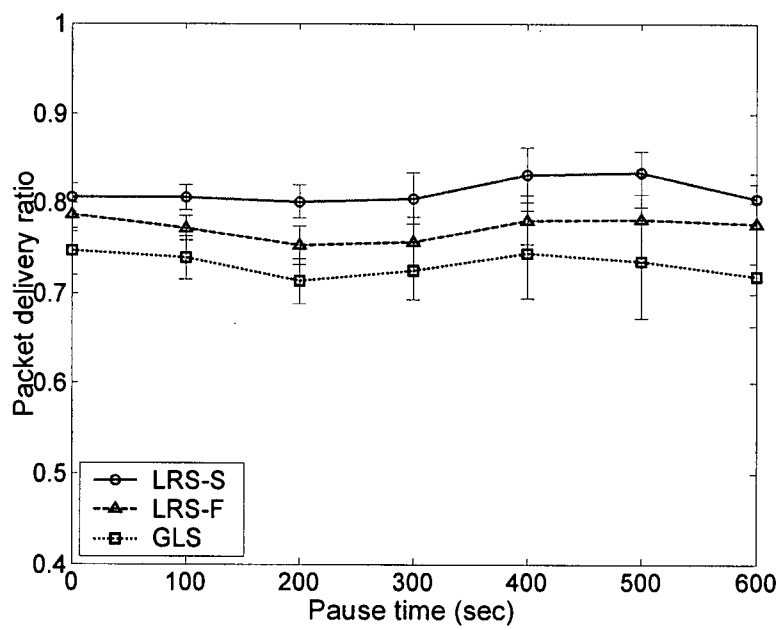



Figure 4.13 Packet delivery ratio between LRSs and GLS over a range of pause time (number of data blackhole attackers = 15).

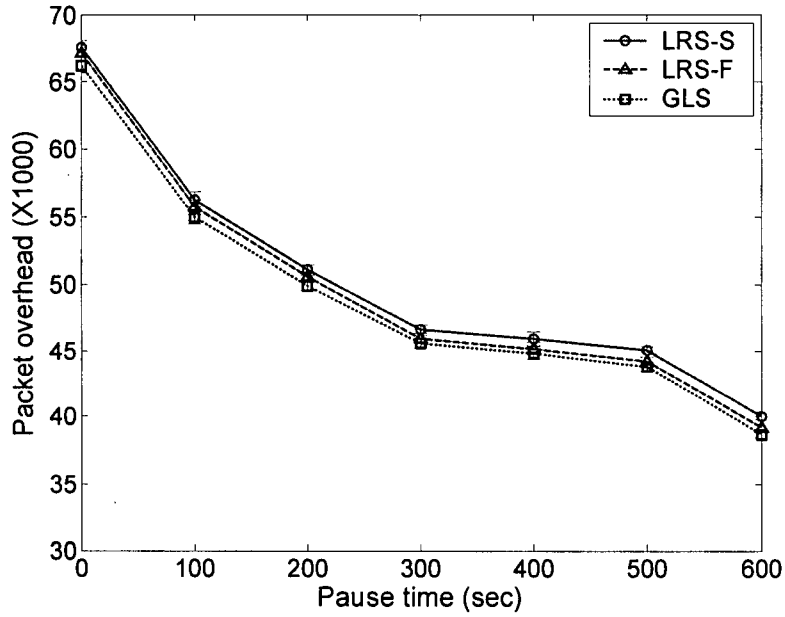



Figure 4.14 Routing packet overhead between LRSs and GLS over a range of pause time (number of data blackhole attackers = 15).

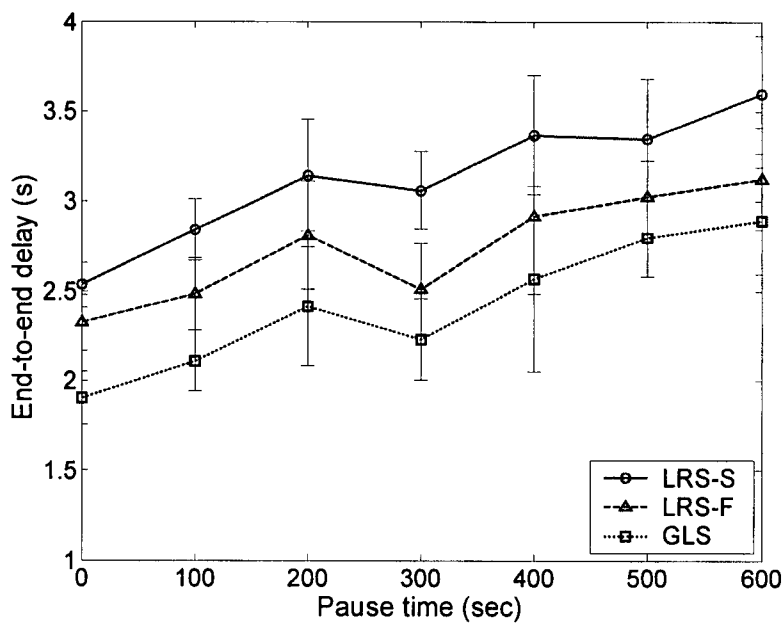



Figure 4.15 Average end-to-end delay between LRSs and GLS over a range of pause time (number of data blackhole attackers = 15).

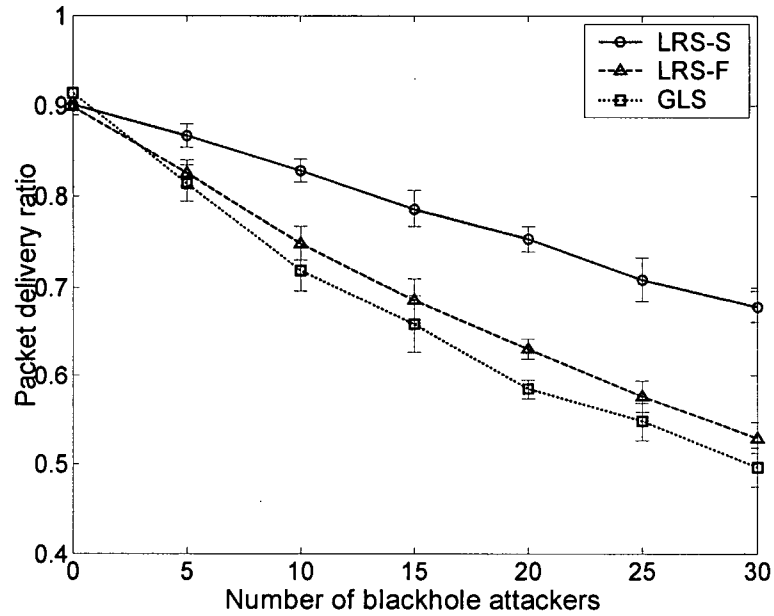



Figure 4.16 Packet delivery ratio between LRSs and GLS with varying number of both data and control blackhole attackers (pause time = 0 sec).

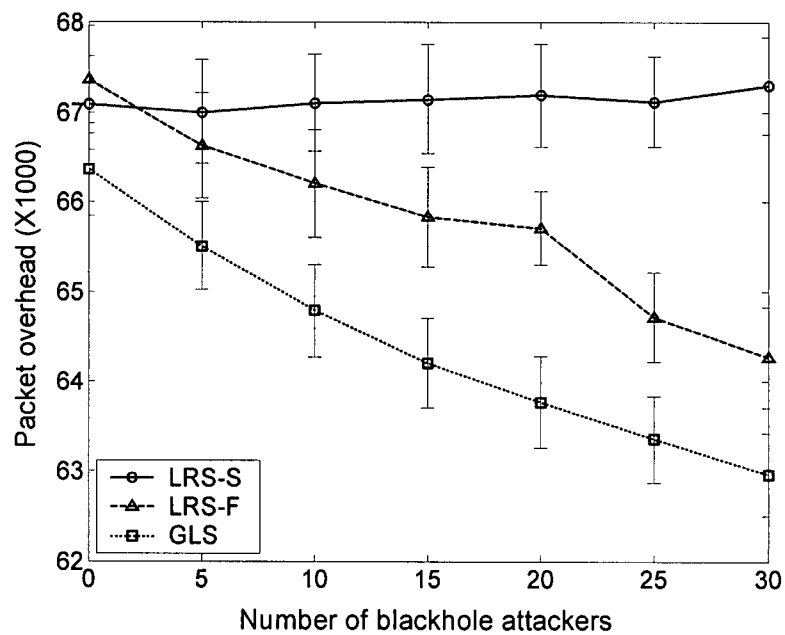



Figure 4.17 Routing packet overhead between LRSs and GLS with varying number of both data and control blackhole attackers (pause time = 0 sec).

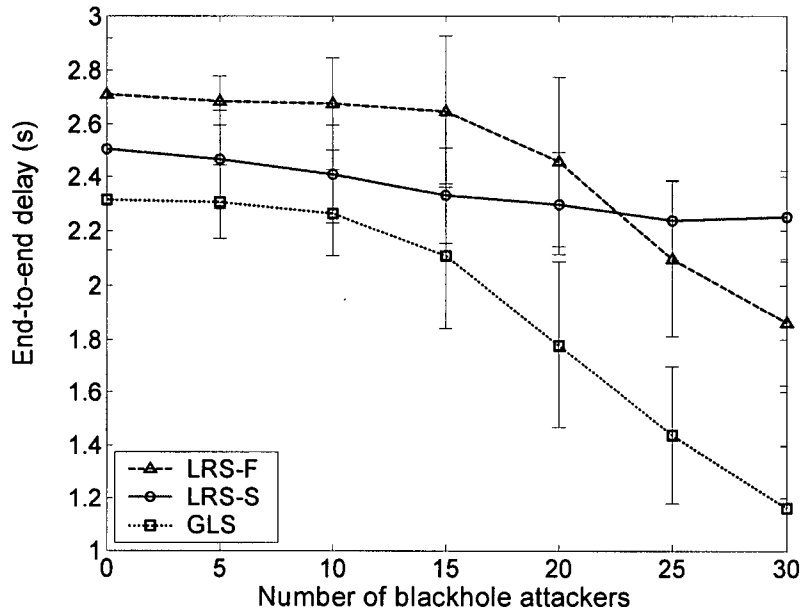



Figure 4.18 Average end-to-end delay between LRSs and GLS with varying number of both data and control blackhole attackers (pause time = 0 sec).

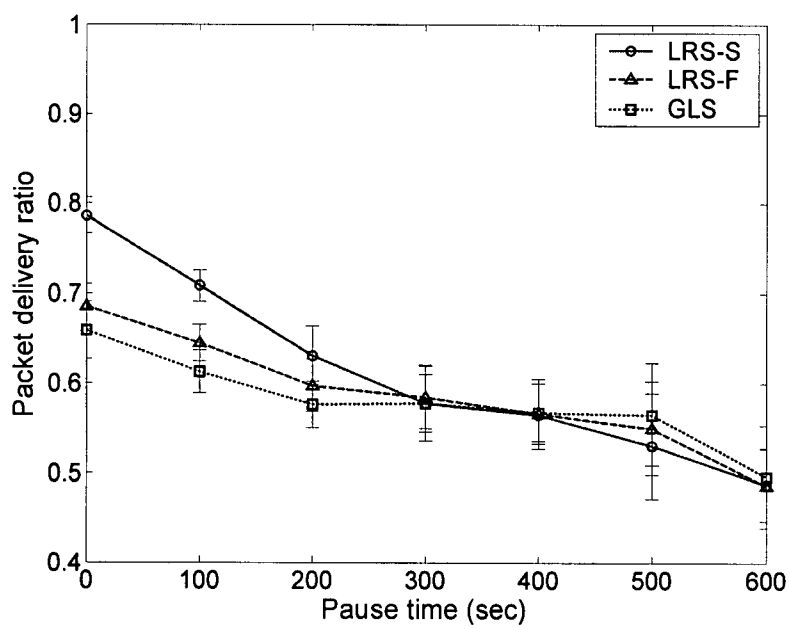



Figure 4.19 Packet delivery ratio between LRSs and GLS over a range of pause time (number of data and control blackhole attackers = 15).

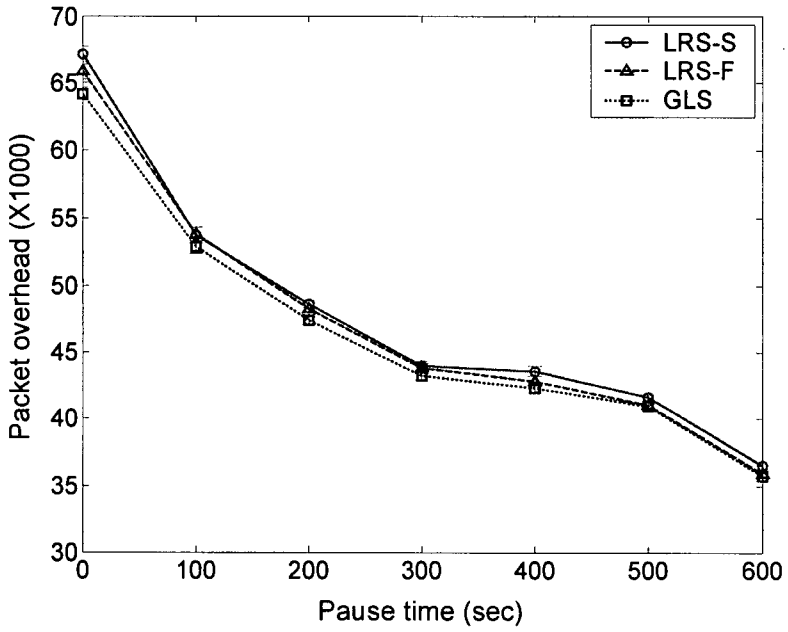



Figure 4.20 Routing packet overhead between LRSs and GLS over a range of pause time (number of data and control blackhole attackers = 15).

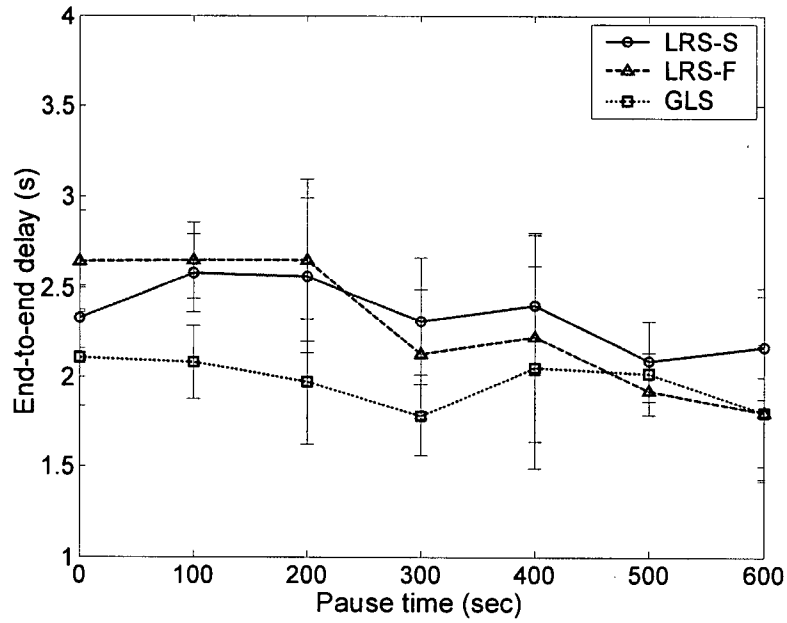



Figure 4.21 Average end-to-end delay between LRSs and GLS over a range of pause time (number of data and control blackhole attackers = 15).

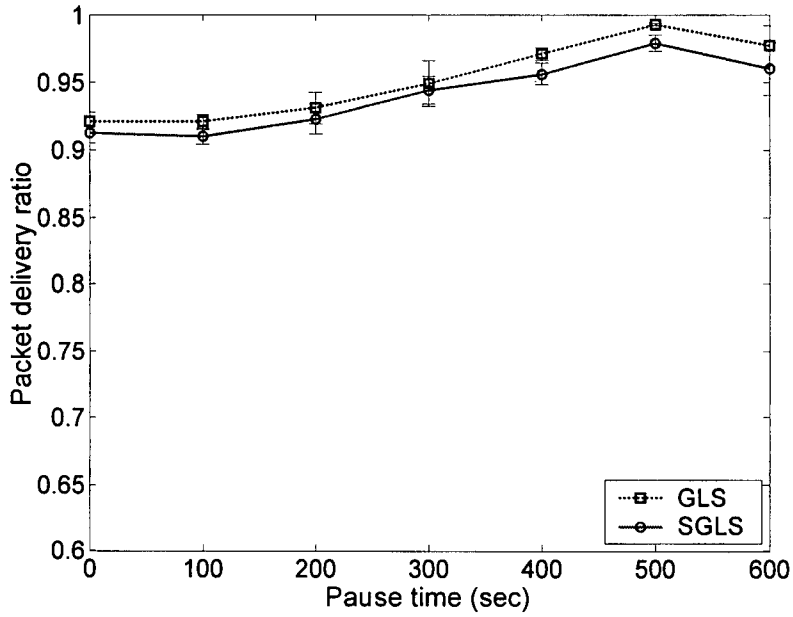



Figure 4.22 Packet delivery ratio between SGLS and GLS over a range of pause time without attackers.

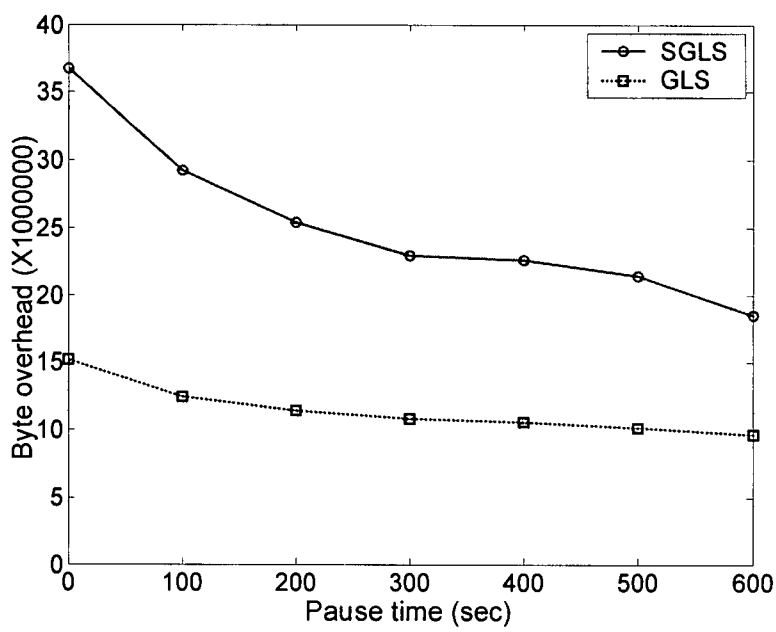



Figure 4.23 Routing byte overhead between SGLS and GLS over a range of pause time without attackers.

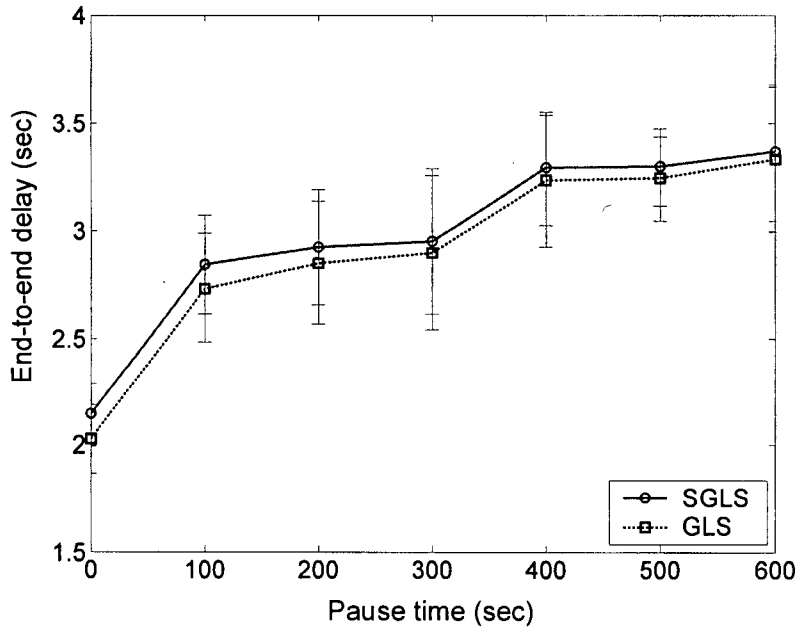



Figure 4.24 Average end-to-end delay between SGLS and GLS over a range of pause time without attackers.

# Bibliography


[1] J.-H. Song, V. Wong, and V. Leung, "A Framework of Secure Location Service for Position-based Ad-hoc Routing," in *Proc. of ACM International Workshop on Performance Evaluation of Wireless Ad Hoc, Sensor, and Ubiquitous Networks* (*PE-WASUN'04*), Venice, Italy, pp. 99-106, Oct. 2004.

[2] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Packet leashes: a defense against wormhole attacks in wireless network," in *Proc. of IEEE Infocom*, San Francisco, CA, Mar./Apr. 2003.

[3] J. Li, J. Jannotti, D.S.J. De Couto, D.R. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. of ACM MobiCom*, Boston, MA, Aug. 2000.

[4] A. Perrig, R. Canetti, D. Song, D. Tygar, and B. Briscoe, "TESLA: multicast source authentication transform introduction," *IETF Internet Draft of Multicast Security Working Group* (work in progress), Aug. 2004.

[5] D. Boneh, G. Durfee, and M. Franklin, "Lower bounds for multicast message authentication," in *Proc. of Eurocrypt, Lecture Notes in Computer Science*, vol. 2045, Springer-Verlag, pp. 437-452, 2001.

[6] C. Madson and R. Glenn, "The use of HMAC-SHA-1-96 within ESP and AH," *IETF RFC* 2404, Nov. 1998.

[7] R. Rivest, "The MD5 message-digest algorithm," *IETF RFC* 1321, Apr. 1992.

[8] Trimble Navigation Limited. *Datasheet & specifications for Trimble Thunderbolt GPS Disciplined Clock.* Sunnyvale, CA. Available at http://trl.trimble.com/ docushare/ dsweb/Get/Document-10015/

[9] IEEE Computer Society LAN/MAN Standards Committee, "Wireless LAN medium access control (MAC) and physical layer (PHY) specifications, IEEE Std. 802.11," Sept. 1999.

[10] M. Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network*, vol. 15, issue 6, pp. 30-39, Nov./Dec. 2001.

[11] G.G. Finn, "Routing and addressing problems in large metropolitan-scale internetworks," Technical Report ISI/RR-87-180, Inst. for Scientific Information, Mar. 1987.

[12] L. Blazevic, L Buttyan, S. Capkun, S. Giordaro, J.-P. Hubaux, and J.-Y. L. Boudec, "Self-organization in mobile ad hoc networks: the approach of terminodes," *IEEE Communications Magazine*, vol. 39, issue 6, pp. 166-174, June 2001.

[13] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. of ACM MobiCom*, Dallas, TX, Oct. 1998.

[14] Z.J. Haas and B. Liang, "Ad hoc mobility management with uniform quorum systems," *IEEE/ACM Transactions on Networking*, vol. 7, no. 2, Apr. 1999.

[15] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *Proc. of Canadian Conference on Computational Geometry*, Vancouver, BC, Aug. 1999.

[16] T.-C. Hou and V.O.K. Li, "Transmission range control in multihop packet radio networks," *IEEE Transactions on Communications*, vol. 34, no. 1, pp. 38-44, Jan. 1986.

[17] S. Marti, T.J. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proc. of ACM MobiCom*, Boston, MA, Aug. 2000.

[18] D.B. Johnson, D.A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks (DSR)," *IETF Internet Draft* (work in progress), July 2004.

[19] P. Michiardi and R. Molva, "CORE: a collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," in *Proc. of IFIP Conference on Security Communications, and Multimedia*, Portoroz, Slovenia, Sept. 2002.

[20] S. Buchegger and J-Y. Le Boudec, "A robust reputation system for mobile ad hoc networks," *EPFL Technical report*, No. IC/2003/50, July 2003.

[21] S. Buchegger and J-Y. Le Boudec, "Performance analysis of the CONFIDANT protocol," in *Proc. of ACM MobiHoc*, Lausanne, Switzerland, June 2002.

[22] L. Zhou and Z.J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, issue 6, Nov./Dec. 1999.

[23] S. Capkun and J.-P. Hubaux, "Secure positioning of wireless devices with application to sensor networks," to appear in *IEEE Infocom* 2005.

[24] Y.-C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: a secure on-demand routing protocol for ad hoc networks," in *Proc. of ACM MobiCom*, Atlanta, GA, Sept. 2002.

[25] P. Papadimitratos and Z.J. Haas, "Secure routing for mobile ad hoc networks," in *Proc. of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference*, San Antonio, TX, Jan. 2002.

[26] M. Zapata and N. Asokan, "Securing ad hoc routing protocols," in *Proc. of ACM Workshop on Wireless Security*, Atlanta, GA, Sept. 2002.

[27] K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields, and E.M. Belding-Royer, "A secure routing protocol for ad hoc networks," in *Proc. of International Conference on Network Protocols*, Paris, France, Nov. 2002.

[28] W. Diffie, P. van Oorschot, and M. Wiener, "Authentication and authenticated key exchanges," *Designs, Codes, and Cryptography*, vol. 2, issue 2, pp. 107-125, June 1992.

[29] B. Schneier, *Applied cryptography: protocols, algorithms, and source code in C*, John Wiley & Sons, Inc., 1996.

[30] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: keyed-hashing for message authentication," *IETF RFC* 2104, Feb. 1997.

[31] D.B. Johnson, "ECC, future resiliency and high security systems," *Certicom White Chapter*, Mar. 1999.

[32] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in *Proc. of ACM Conference on Computer and Communications Security*, Philadelphia, PA, Nov. 2001.

[33] Helion Technology Ltd. *High performance solutions in silicon – MD5 core*. Available at http://www.heliontech.com/core5.htm

[34] NS-2 for grid. Available at http://www.pdos.lcs.mit.edu/grid/sim/index.html

[35] M. Brown, D. Cheung, D. Hankerson, J.L. Hernandez, M. Kirkup, and A. Menezes, "PGP in constrained wireless devices," in *Proc. of the USENIX Security Symposium*, Denver, CO, Aug. 2000.

# Chapter 5 Conclusions

We conclude this dissertation with a summary of our contributions and directions for future work.

## 5.1 Summary

The first part of this research began with a study of load-balancing routing protocol in mobile ad hoc wireless access network.

- When flooding-based on-demand route discovery is used in mobile ad hoc wireless access networks, many routing messages (e.g., RREQ) may be propagated unnecessarily. In Chapter 2, we proposed an extension of the ad hoc on-demand routing protocol by incorporating the concept of load-balancing. Our proposed LB-AODV protocol is simple and well-suited for the mobile ad hoc wireless access network environment. We compared the performance of our proposed LB-AODV protocol with both the original AODV and gossip-based routing protocols in different mobility and traffic scenarios. Simulation results show that LB-AODV delivers more data packets to the gateway and decreases the end-to-end delay of packets delivered by reducing the transmissions of routing control messages by 50 % or more. In scenarios with traffic congestion, LB-AODV significantly outperforms AODV and GOSSIP1 routing protocols. We compared the performance of the protocols in a scenario with a larger number of mobile nodes accessing two gateways. LB-AODV provides significant advantages over AODV and GOSSIP1 in terms of throughput and routing overhead even in a large network with two gateways. Although we presented the details of LB-AODV based on the AODV routing protocol, the load-balancing concept developed in this chapter can generally be applied to other on-demand routing schemes. Recently, there has been a lot of interest in deploying the wireless mesh

networks (e.g., [1]-[5]). Our work on load balancing can be deployed in those networks which use AODV routing protocol to improve the network performance.

The second part of this thesis focused on the design of secure topology-based and position-based routing protocols in MANET.

- In Chapter 3, we proposed the Tamper Resistant Module (TRM) and the Secure Table Entry Protection (STEP) mechanism to prevent routing table tampering attacks. STEP provides the authentication for both the destination sequence number and hop-count fields in the routing table entry. The receiving node can confirm the correctness of message by verifying the signatures of two consecutive upstream nodes and the source node. We described how STEP can be incorporated in the AODV routing protocol. We proposed the Efficient STEP (ESTEP), which can avoid the use of expensive multiple digital signatures on RREQ broadcast messages. We also identified the security threats and analyzed the security requirements for AODV routing protocol in mobile ad hoc networks. In light of these analyses, we proposed a secure routing extension (SeAODV) and Secure Data Forwarding (SDF) mechanism for the AODV routing protocol. SeAODV uses digital signatures in both RREQ and RREP packets to prevent malicious users from joining a valid path from source to destination. Also, both RREQ and RERR packets are extended to avoid suspicious links. For secure data transmission, we introduced the use of HMAC to maintain the integrity of data messages. Simulation results showed that both STEP and ESTEP continue to maintain a high packet delivery fraction and a small end-to-end delay at the expense of slightly higher route acquisition latency and control overhead in route discovery. In the presence of either data packet dropping or routing table tampering attacks, SeAODV continues to maintain a high packet delivery fraction and a small end-to-end delay. As different kinds of mobile devices continue to proliferate, it is expected that more

mobile devices will increase the use of the ad hoc mode in different applications (e.g., songs or video files sharing, games) in near future. Our work on secure routing protocols can prevent different potential attacks in the network layer.

- In Chapter 4, we identified the security threats and analyzed the security requirements for position-based routing protocols in mobile ad hoc networks. In consideration of these requirements, we proposed a Secure Geographic Forwarding (SGF) mechanism which can provide *message authentication*. By combining SGF with the Grid Location Service (GLS), we proposed a Secure Grid Location Service (SGLS) where any receiver can verify the correctness of location messages. We also proposed a Local Reputation System (LRS) aiming at detecting and isolating misbehaving neighboring nodes. The security mechanisms added to GLS include TIK, TESLA, MAC, digital signature, and a reputation system. Simulation results showed that in the presence of message dropping attacks, the proposed LRS mechanism maintains a high message delivery ratio at the expense of a higher average end-to-end delay and routing overhead in general. We have also investigated the computational complexity of SGLS through analysis and simulations.

## 5.2 Further Work

In the course of the investigations reported in this thesis, a number of interesting problems have been discovered which merit further study.

- **Load-balancing Routing**: To facilitate practical implementation of load-balancing scheme, we need to investigate techniques that can provide good estimations of network size and topology in a dynamic MANET. It is also necessary to seek further improvements of group assignment mechanism, especially for large networks with multiple gateways. Furthermore, it may be interesting to investigate how the load-

balancing concept can be incorporated in other on-demand routing protocols with different routing metrics (e.g., the least load or the least power route).

- **Key Management in MANET**: Most of secure routing protocols require the use of some kind of cryptographic keys between two communicating nodes. The dissemination of authentic keys is still an open management problem. Researchers have proposed several distributed solutions by using either threshold cryptography [6] or the chains of trust [7] in MANETs. However, the threshold cryptography is not suitable for highly partitioned networks, and the chains of trust by accepting all other users' certificates is not desirable in some applications. To design a robust key management system, it is crucial to: (1) provide high service availability in highly partitioned networks; (2) avoid the transitivity of trust; (3) support minimal pre-configuration during the network deployment phase; and (4) to handle the joining and leaving of nodes in the networks.
- **Open Challenges for Security in MANET**: The research of secure MANET is in its early stage. There is still room to come up with and analyze other feasible attacks. Such analysis may enable researchers to verify protocol security by using formal methods. Another challenge is to design efficient protocols that can provide both strong security and high network performance. However, when more security features are introduced into the network, there is an increase in computation and communication overhead. Further work is required on the study of the trade-offs between the strength of security mechanisms and the network performance.

# Bibliography


[1] Motorola Inc; http://www.motorola.com/

[2] Radiant Networks; http://www.radiantnetworks.com

[3] Strix Networks; http://www.strixsystems.com

[4] Tropos Networks; http://www.tropos.com

[5] Mesh Dynamics; http://www.meshdynamics.com

[6] L. Zhou and Z.J. Haas, "Securing ad hoc networks," *IEEE Network*, vol. 13, issue 6, Nov./Dec. 1999.

[7] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *Proc. of ACM MobiHoc*, Long Beach, CA, Oct. 2001.