A SECURITY FRAMEWORK FOR
MOBILE AD HOC NETWORKS

A Dissertation in
Computer Science and Engineering
by
Hung-Yuan Hsu

© 2008 Hung-Yuan Hsu

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Doctor of Philosophy

December 2008
The dissertation of Hung-Yuan Hsu was reviewed and approved* by the following:

Ali R. Hurson  
Professor of Computer Science and Engineering  
Dissertation Co-Advisor  
Co-Chair of Committee

Sencun Zhu  
Assistant Professor of Computer Science and Engineering  
Dissertation Co-Advisor  
Co-Chair of Committee

Trent Jaeger  
Associate Professor of Computer Science and Engineering

Piotr Berman  
Associate Professor of Computer Science and Engineering

Ling Rothrock  
Associate Professor of Industrial and Manufacturing Engineering

Mahmut Kandemir  
Associate Professor of Computer Science and Engineering  
Graduate Officer for the Department of Computer Science and Engineering

*Signatures are on file in the Graduate School
ABSTRACT

Mobile Ad Hoc Networks are networks composed of autonomous resource scarce mobile nodes, built upon open media accessible to anonymous nodes, and operating in an isolated environment where each node has to defend itself without trustworthy gateways. These characteristics make the security issues more challenging in a Mobile Ad Hoc Networks than the Internet, since a wide variety of attacks could exploit its weaknesses. In this thesis, we propose a security framework for Mobile Ad Hoc networks which consists of a network access control protocol and an online traceback protocol. The thesis addresses five major challenges in mobile ad hoc networks security. It:

(i) Introduces a lightweight mechanism to restrict non-authorized network access,

(ii) Proposes a theoretical model to study the characteristics of the online traceback,

(iii) Proposes a protocol to localize the attack,

(iv) Studies the impact of performance by introducing the aforementioned network access control and online traceback protocols, finally,

(v) Evaluates the power efficiency of the proposed security framework.

As advanced in the literature, most ad hoc networks do not implement any network access control, leaving these networks vulnerable to the attack by malicious party injecting packets into the network with the goal of depleting the resources. A Lightweight Inter-layer Protocol (LIP) is proposed to prevent packet injection attacks based on an efficient local broadcast authentication mechanism. Our research makes the first effort to quantitatively analyze the impacts of node mobility, attack packet rate, and intrusion response time on the traceability of two types of well-known IP traceback schemes: namely, probabilistic packet marking (PPM) and
hash-based logging. Based on the principle of divide and conquer, we propose an efficient online traceback scheme that works by dividing a forwarding path into multiple interweaving fragments. We assess the performance of the proposed security protocols by conducting extensive simulations studies. Finally, the thesis introduces a power consumption model and evaluates the power efficiency of the proposed security framework.
# TABLE OF CONTENTS

LIST OF FIGURES ................................................................................................................. viii

LIST OF TABLES ................................................................................................................... xi

LIST OF ABBREVIATIONS .................................................................................................. xii

ACKNOWLEDGEMENTS ..................................................................................................... xiii

Chapter 1 Introduction ......................................................................................................... 1

  1.1 Statement of the Problem and Its Importance ..................................................... 2
  1.2 Research Objectives and Contributions ........................................................... 5
    1.2.1 Network Access Control ................................................................. 5
    1.2.2 Traceback ......................................................................................... 6
    1.2.3 Power Efficiency of the MANETs Security Framework ....................... 7
  1.3 Dissertation Outline .......................................................................................... 8

Chapter 2 Background ....................................................................................................... 10

  2.1 Network Access Control .................................................................................. 10
    2.1.1 TESLA .............................................................................................. 15
    2.1.2 A Lightweight Inter-layer Protocol (LIP) ........................................... 17
  2.2 Traceback ......................................................................................................... 19
    2.2.1 IP Traceback .................................................................................... 19
    2.2.2 Traceback in MANETs ..................................................................... 26
  2.3 Power Efficiency ............................................................................................... 28
    2.3.1 Power Consumption Model ............................................................. 29
    2.3.2 Wake-up based power management ................................................ 32
  2.4 Conclusion ......................................................................................................... 38

Chapter 3 Network Access Control ................................................................................ 40

  3.1 Introduction ....................................................................................................... 40
  3.2 Assumptions and Design Goal ......................................................................... 42
    3.2.1 Security Assumptions ....................................................................... 42
    3.2.2 Security Assumptions ....................................................................... 43
    3.2.3 Design Goal ...................................................................................... 44
  3.3 Overview ........................................................................................................... 45
  3.4 Scheme I: Basic Scheme .................................................................................. 47
    3.4.1 Using One-time Authentication Keys ............................................. 47
    3.4.2 Random Neighborship Verification .............................................. 50
  3.5 Scheme II: a location-aware verification scheme ............................................ 52
  3.6 Further discussions ......................................................................................... 57
    3.6.1 Interaction with routing protocols ................................................. 57
    3.6.2 Key chain generation and renew .................................................. 58
    3.6.3 Legacy issue .................................................................................... 58
5.4.2 Communicational energy analysis .................................................................128
5.4.3 Simulation results ..........................................................................................130
5.5 Conclusion ..........................................................................................................135

Chapter 6 Concluding Remarks and Future Work ..............................................137

6.1 Contributions of this Thesis ...........................................................................137
6.2 Future Research .................................................................................................139

Bibliography .............................................................................................................141
LIST OF FIGURES

Figure 2-1: The steps that an adversary forms a loop by spoofing of packets. ..................12

Figure 2-2: The sequence of secure routing message exchange in ARAN. ...........13

Figure 2-3: A scenario in LHAP when node A joins the network. B, C, D, E are its current neighbors. The graph is the current network topology. ...............15

Figure 2-4: TESLA key chain. ..................................................................................16

Figure 2-5: TESLA packets authentication. ..............................................................17

Figure 2-6: Controlled Flooding. .............................................................................19

Figure 2-7: ICMP Traceback. .................................................................................21

Figure 2-8: Probabilistic Packet Marking (PPM). ..................................................23

Figure 2-9: Hash-based Packet Logging Traceback. .............................................24

Figure 2-10: Power consumption of a mobile device iPAQ. ...............................28

Figure 2-11: An example that shows two neighboring node can listen to each other with clock drifts .................................................................34

Figure 2-12: Quorum based wake-up schedule. The filled cells indicate the beacon intervals at which the node wakes up. ..............................................35

Figure 2-13: A (7,3,1) WSF design where 3 slots are ON slots out of a 7-slot cycle. ..35

Figure 3-1: The protocol stack in which LIP is between the network layer and the data link layer. .................................................................46

Figure 3-2: A Triangle inequality. ...........................................................................50

Figure 3-3: A snapshot of nodes u and v moving at velocities $\vec{V}_u$ and $\vec{V}_v$, respectively. Dashed circles indicate the transmission range of the two nodes. The analysis is used to predict when the two nodes will move out of the transmission range of each other....53

Figure 3-4: Node v’s state transition diagram. Each neighbor u belongs to one of the four states. ..............................................................................56

Figure 3-5: Impact of node mobility on control overhead. .....................................67

Figure 3-6: Impact of node mobility on traffic delivery ratio. ..............................68
Figure 3-7: Impact of node mobility on the length of buffer .................................................. 69
Figure 3-8: Impact of node density on control overhead (node mobility 10 m/s). ................. 70
Figure 3-9: Impact of node density on traffic delivery ratio (node mobility 10 m/s). ............. 71
Figure 3-10: Impact of location awareness on control overhead (p = 10%, node speed 10 m/s). .......................................................................................................................... 72
Figure 3-11: Control overhead under different mobility models ....................................... 74
Figure 3-12: REDs under different mobility models .......................................................... 74
Figure 4-1: An attack path of packet M where node S and node b3 are the compromised nodes. S injects bogus packets, and b3 conspires to neutralize the traceback attempt. ... 80
Figure 4-2: The traceability \( \mathbb{S}_{ppm} \) of PPM traceback in MANETs versus path length \( d \) with varying packet rates. ......................................................................................... 85
Figure 4-3: The traceback latency. \( t_{SV} \) is the end-to-end propagation delay from \( u_1 \) to \( V \). \( t_{SV} \) is the end-to-end delay from \( V \) to \( u_1 \). \( T_{resp} \) is the victim node response time .......... 87
Figure 4-4: Traceability as the function of path length. The dashed lines present the theoretical traceability and the solid lines present the simulation results. \( T_{resp} \) denotes the response time of the victim node ......................................................................... 89
Figure 4-5: The maximum path length under the constraint of the traffic delivery ratio ...... 91
Figure 4-6: An example of packet marking where \( u_1 \) and \( u_3 \) are markers in regard to \( M_1 \), and \( u_2 \) and \( u_5 \) are markers in regard to \( M_2 \). ................................................................. 93
Figure 4-7: The distance-based marking probability ............................................................ 98
Figure 4-8: The flowchart a victim \( V \) follows to process a received traceback report ....... 100
Figure 4-9: Hotspot size as a function of the IDS response time ......................................... 106
Figure 4-10: Hotspot size as a function of network mobility .............................................. 107
Figure 4-11: Hotspot size versus IDS response time in a realistic mobility model ............ 107
Figure 4-12: The benefit of virtual links. (The pkt# denotes the number of packet digests embedded in a traceback query) ............................................................................... 108
Figure 4-13: Comparison with logging-based traceback on hotspot size ............................. 109
Figure 5-1: The impact of the node mobility on the computational power consumption .... 122
Figure 5-2: The impact of the node mobility on the communicational power consumption. ................................................................. 122

Figure 5-3: The impact of the node mobility on the computational power consumption. ....... 123

Figure 5-4: The impact of the location awareness of the nodes. ........................................... 124

Figure 5-5: The communicational overhead of LIP versus LHAP. ........................................ 125

Figure 5-6: For each traceback query, the query message is embedded with malicious packet digests varying from 1 to 10 where 1 represents the victim node launches a corresponding request for each received malicious data packet. ........................................... 131

Figure 5-7: The power consumption and the hotspot size versus the packet rate from the attack source. ........................................................................................................................................... 133

Figure 5-8: The power consumption and the hotspot size versus the node density. ............ 134

Figure 5-9: The communicational power overhead of online multicast versus flooding traceback protocols. ........................................................................................................................................... 135
LIST OF TABLES

Table 1-1: Attacks in link and network layers. .................................................................3
Table 2-1: Comparative analysis of access control schemes. .............................................18
Table 2-2: Comparative analysis of IP traceback schemes.................................................25
Table 2-3: Comparison of traceback schemes in MANETs.................................................27
Table 2-4: Lucent IEEE 802.11 WaveLAN PC CARD (2 Mbps). ........................................30
Table 2-5: Power Consumption Model. .............................................................................32
Table 2-6: Comparative analysis of different power saving schemes...............................38
Table 3-1: Simulation parameters. .....................................................................................65
Table 3-2: The mobility model RPGM parameters............................................................73
Table 3-3: The mobility model Manhattan parameters. ......................................................73
Table 4-1: Simulation parameters. .....................................................................................104
Table 5-1: Computation time and energy consumption for 128-bit multiply result. ............113
Table 5-2: Energy Consumption Estimates for digital signature. .........................................113
Table 5-3: AES computational energy consumption estimate. ..........................................114
Table 5-4: Energy consumption estimates. ........................................................................114
Table 5-5: The estimation of the computational power of packet authentication................116
Table 5-6: The estimation of the computational power of neighborship verification............117
Table 5-7: The estimation of the communicational power of packet authentication.............118
Table 5-8: The estimation of the communicational power of neighborship verification.......119
Table 5-9: The estimation of the communicational power of neighborship verification.......120
Table 5-10: Simulation parameters. ..................................................................................121
Table 5-11: The estimation of the computational power of packet marking. ......................128
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BSS</td>
<td>Base Station</td>
<td>25</td>
</tr>
<tr>
<td>DDoS</td>
<td>Distributed Denial of Service</td>
<td>19</td>
</tr>
<tr>
<td>DGAs</td>
<td>Data Generation Agents</td>
<td>23</td>
</tr>
<tr>
<td>DoS</td>
<td>Denial of Service</td>
<td>3</td>
</tr>
<tr>
<td>DPM</td>
<td>Deterministic Packet Marking</td>
<td>21</td>
</tr>
<tr>
<td>ISPs</td>
<td>Internet Service Providers</td>
<td>7</td>
</tr>
<tr>
<td>LHAP</td>
<td>Lightweight Hop-by-Hop Authentication Protocol</td>
<td>14</td>
</tr>
<tr>
<td>LIP</td>
<td>Lightweight Inter-layer Protocol</td>
<td>6</td>
</tr>
<tr>
<td>MACs</td>
<td>Message Authentication Codes</td>
<td>13</td>
</tr>
<tr>
<td>MANETs</td>
<td>Mobile Ad Hoc Networks</td>
<td>1</td>
</tr>
<tr>
<td>NIC</td>
<td>Network Interface Card</td>
<td>13</td>
</tr>
<tr>
<td>PPM</td>
<td>Probabilistic Packet Marking</td>
<td>6</td>
</tr>
<tr>
<td>QoS</td>
<td>Quality of Service</td>
<td>2</td>
</tr>
<tr>
<td>RFID</td>
<td>Radio Frequency Identifier</td>
<td>34</td>
</tr>
<tr>
<td>RREP</td>
<td>Route Reply</td>
<td>11</td>
</tr>
<tr>
<td>RREQ</td>
<td>Route Request</td>
<td>11</td>
</tr>
<tr>
<td>SPIE</td>
<td>Source Path Isolation Engine</td>
<td>22</td>
</tr>
<tr>
<td>TBF</td>
<td>Tagged Bloom Filter</td>
<td>24</td>
</tr>
<tr>
<td>TESLA</td>
<td>Timed Efficient Stream Loss-tolerant Authentication</td>
<td>15</td>
</tr>
<tr>
<td>TTL</td>
<td>Time To Live</td>
<td>20</td>
</tr>
<tr>
<td>WSF</td>
<td>Wakeup Schedule Function</td>
<td>32</td>
</tr>
</tbody>
</table>
ACKNOWLEDGEMENTS

First of all, I would like to acknowledge my two beloved co-advisors Dr. Ali R. Hurson and Dr. Sencun Zhu for their wonderful guidance through my entire Ph.D discipline. Through this process, I am becoming more mature in doing research. I also would like to thank my committee Dr. Trent Jaegar, Dr. Piotr Berman, and Dr. Ling Rothrock for helping to shape this work. Lastly, I would like to thank the Department of Computer Science and Engineering, especially the graduate staff Karen Corl and IT staffs, and the staff in the Graduate School for helping to arrange everything.

I would like to thank the U.S. government agencies that have provided financial support for my work. This work in part has been supported by the National Science Foundation under contract IIS-0324835.

Finally, my greatest thanks go to my parents and my sister. Their unconditional support and love have accompanied me through the entire Ph.D. process. I cannot achieve what I have achieved without them.

God bless you all
To my parents
Chapter 1

Introduction

Mobile ad hoc networks (MANETs) consist of a group of mobile nodes which are connected by wireless communication. The topology of a MANET varies from time to time without prior notice. Due to the absence of a defined infrastructure, and mobility, MANETs are dynamic and heterogeneous environment. MANETs provide unique advantages over structured wireless networks under certain scenarios; the communication and surveillance in a battle field, the emergency communication in a disaster scene, and home or office networking in a small area, e.g. a conference room, and a building [1]. MANETs possess many prominent features outstanding the present mobile wireless networks that rely on the support of fixed network infrastructure. First, with self-creating, self-organizing, and self-administering properties, MANETs can be quickly deployed to almost arbitrary environments with a low installation cost. Second, without sticking to a specific routing protocol and single point of failure, routing in MANETs is more flexible and intrinsically fault resilient.

Nonetheless, these benefits come at the expense of several challenges especially in the aspect of security threats. The fact that nodes participate in network protocols as peers basically implies that any individual network node can abuse protocol operations. Without a priori trust relationship between the nodes in MANETs, network nodes cannot be trusted to execute network functionalities as expected by designers [6]. Furthermore, the shared wireless media accessible to both legitimate network members and malicious attackers exposes vulnerabilities to adversaries who can make use of it to launch numerous proactive or passive attacks, e.g. eavesdropping, impersonating, or injecting malicious data. Compromised mobile nodes can even launch attacks from within the network.
Consequently, while early MANETs research efforts concentrated on functional issues, e.g. routing protocols, power conservation, Quality of Service (QoS), etc. [1][7][8], in recent years, MANETs security has drawn tremendous attention [1][2]. MANETs security threats countermeasures can be classified into two approaches: proactive and reactive [2]. The proactive approach is to forefend attackers from launching attacks at the very beginning. Generally, it is achieved through applying cryptographic techniques [21][31]. On the other hand, the reactive approach attempts to react according to the detected attacks afterward. A threat can be detected in neighboring or in an end-to-end manner. The misbehavior of malicious nodes can be detected directly by neighboring nodes through overhearing the communication channel [14][30][52]. The multi-hop attack can be caught by the destination node using intrusion detection techniques [42]. On the alert of ongoing attacks, neighboring nodes can isolate the malicious node through the operations ranging from dropping its packets or blacklisting it in route selection. As to the destination receiver, it can report to an administrative authority in a managed environment [6] or stage an attack source traceback, which will be studied in this dissertation.

Finally, MANETs participants generally rely on battery power; due to the slowly advancing battery technology, energy continues to be a precious resource in MANETs. The proposed security measures should only impose a limited amount of energy overhead to MANETs.

1.1 Statement of the Problem and Its Importance

The research goal is to provide an integrated security solution to ensure the protection of data delivery from one node to another node within a MANET. The multi-hop connectivity of MANETs is provided by network layer and data link layer. Hence, this research considers the
security threats and the corresponding countermeasures spanning these two layers. The attacks in data link layer and network layer of MANETs can be summarized as follows (see Table 1-1) [3].

Table 1-1: Attacks in link and network layers.

<table>
<thead>
<tr>
<th>Layer</th>
<th>Single Layer Attacks</th>
<th>Multiple Layers Attacks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Link</td>
<td>IEEE 802.11 MAC transmission attack.</td>
<td>Impersonation attack.</td>
</tr>
<tr>
<td></td>
<td>IEEE 802.11 WEP cryptography attacks.</td>
<td>Denial of Service (DoS) attack.</td>
</tr>
<tr>
<td></td>
<td>Wormhole attack.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Byzantine attack.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Rushing attack.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Resource consumption attack.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Location disclosure attack.</td>
<td></td>
</tr>
</tbody>
</table>

In addition to the attacks launched in a particular layer, sophisticated attacks can target multiple layers; i.e., impersonation attack, denial of service (DoS) attacks, and man-in-the-middle attacks [3]. Currently, there have been countermeasures proposed for the well known security threats [1][2]. However, many of these solutions are restrictive and designed as a countermeasure for a well known attack model or for a specific routing protocol [19][20][39]. The major shortcoming of defending security threats in this manner is the lack of a full aspects consideration. On the one hand, as noted in [2][9], the overall security of a multi-fence security solution is as secure as the weakest link in the system, just like a chain. Missing a single component or failing to seamlessly integrate all different parts may dramatically downgrade the overall security level. On the other hand, network performance is as important as security strength. Security mechanisms always consume resource which is precious in MANETs. To balance the two dimensions is a challenge in MANETs security design. A comprehensive solution
should use resources more efficiently rather than just putting different mechanisms together. Moreover, different attacks may be mutually related. For instance, an impersonation attack is generally the first step for other more sophisticated attacks. If we can defend this attack, further attacks can be prevented as well.

Consequently, this research approaches the security as an integrated defense solution rather than considering a specific network protocol. It considers the following security challenges that have not been well addressed in the literature:

- **Susceptible Network Link**: Wireless links between nodes are highly vulnerable to link attacks, which include passive eavesdropping, active interfering, data tampering, impersonation, message replay, message distortion, and denial of service. Active attacks might allow the adversary to delete messages, to inject malicious/erroneous messages, to modify messages, and to impersonate nodes, thus violating availability, integrity, authentication, and nonrepudiation.

- **Compromised Nodes**: MANETs may be deployed in a hostile environment such as a battle field. Network nodes may be physically captured or compromised. Therefore, internal attacks are realities to face in MANETs. Internal attacks, also referred to as insider attacks [1], are launched by compromised network members. Internal attacks are more severe in comparison to external (outsider) attacks because insiders possess much valuable secret information, and cannot be hindered by network access control mechanisms. Some of the internal attacks can be detected by monitoring neighbor node misbehaviors, but a primary challenge is to identify the adversary staging the attack multi-hops away from the victim node.

- **Lack of a Priori Trust**: The correct functioning of MANETs critically relies on the cooperation of network nodes. Without the trusted relationship between network nodes, classical network security mechanisms cannot deal with selfish or malicious nodes. For
instance, the proposed Internet Protocol traceback (IP traceback) schemes heavily rely on the trustworthiness of the information provided by intermediate routers.

The aforementioned challenges incite the following research topics across the link layer and the network layer independent of any particular protocol.

1.2 Research Objectives and Contributions

Within the scope of the ad hoc networks, this research is intended to address the issues of network access control, attack source traceback, and the power overhead of the proposed security measures.

- **Network Access Control**: The research introduces a lightweight network access control protocol, which employs authentication mechanisms to assure only authorized nodes may send packets into the network.

- **Attack Source Traceback**: The research analyzes the characteristics of the online traceback in ad hoc networks, and proposes a feasible protocol in an attempt to trace the attack source or the area where the malicious nodes reside (*hotspot*).

- **Power Efficiency of the MANETs Security Framework**: The research explores the power overhead incurred by the proposed MANETs security framework through an analysis based on a power consumption model and the simulation results.

1.2.1 Network Access Control

In packet routing, forwarding packets to well-behaving nodes and denying accesses from misbehaving nodes are critical for the proper functioning of a mobile ad hoc network where the cooperation among all network nodes is usually assumed. However, the lack of a network
infrastructure, the dynamic of the network topology and nodes membership, and the potential internal attacks launched by malicious, compromised, and/or non-cooperative selfish nodes make the conventional network access control mechanisms not applicable. Specifically, the lack of network access control leaves the network vulnerable to packet injection attacks where a malicious node injects packets into the network with the goal of depleting the resources of the nodes relaying on the packets. In this research, we design a Lightweight Inter-layer Protocol (LIP) [94] for preventing packet injection attacks based on an efficient local broadcast authentication mechanism. The protocol employs efficient authentication mechanisms, ensuring that only authorized nodes can inject traffic into the network. In addition, it can detect and minimize the impersonation attacks by compromised nodes. Moreover, the transparency and independence of LIP allows it to be turned on/off as desired and to be integrated seamlessly with secure routing protocols, providing stronger security services for ad hoc networks. Through detailed simulation study, we show that the protocol is scalable, and it incurs small bandwidth overhead as well as little impact on the traffic delivery ratio even in the case of high node mobility.

1.2.2 Traceback

IP traceback in Internet has well been studied for years [72]. However, the existing IP traceback techniques assume reliable intermediate routers and relative static network topology (e.g., Internet). Hence, they are not applicable to MANETs. Recently, a few traceback techniques for MANETs have been proposed in the literature [58]. The challenges posed by MANETs are two folds; first, the dynamic topology of a MANET makes many classical IP traceback schemes inapplicable. For example, it becomes futile, in MANETs, to reconstruct upstream attack paths by employing probabilistic packet marking (PPM) because packet routes in MANETs change from
time to time. Moreover, the lack of boundary in such a peer-to-peer open network architecture makes it impossible to identify the *entry point* of the attack in the local domain. Second, the Internet routers are assumed to be reliable because most of the current Internet routers are managed by Internet Service Providers (ISPs). Although it is still debatable if there are enough incentives to have ISPs involve in tracking anonymous attacks [47], nevertheless, the information provided by ISPs is assumed to be trustworthy. However, the intermediate nodes in MANETs are not credible because the constituents of MANETs tend to be compromised if a MANET is deployed in a hostile environment. In this research, we first quantitatively analyze the impacts of node mobility, attack packet rate, and intrusion response time on the traceability of two types of well-known IP traceback schemes: *probabilistic packet marking* (PPM) and *hash-based logging*. Then we propose a hybrid online traceback scheme that integrates the advantages of both PPM and logging schemes. Based on the principle of divide and conquer, it works by dividing a forwarding path into multiple smaller interweaving fragments. A PPM-like scheme is used to reconstruct these fragments and a logging-like scheme is applied to gather fragmentation information during traceback. Through simulation we show that the hybrid scheme could effectively trace an attack in terms of a small *hotspot*, where the attacker resides, under various settings.

1.2.3 Power Efficiency of the MANETs Security Framework

Most of the mobile devices making up the MANETs rely on the battery power. Due to the limited battery capacity and the slowly advancing battery technology, energy continues to be a precious resource to MANETs. In the past few years, energy consumption behavior of wireless devices [95][96], power management in ad hoc networks [97][98][100], and power-aware ad hoc routing protocols [99][101][102][103] have been widely studied in the literature. On one hand,
power-awareness design is critical to the success of any ad hoc network protocol. On the other hand, security mechanisms generally come at the expense of consuming the system resources. Therefore, this research comprehensively considers the power overhead incurred by deploying the proposed security framework. For example, in the network access control protocol, the neighboring network nodes are required to periodically exchange credentials with each other so as to refresh the trustworthy neighbor ships; it implies that the mobile devices turn on the wireless network interface more often to overhear the traffic, receive/transmit the control messages, and retransmit the data packets due to the higher chance of media collision and packets congestion. The goal of this research is to show the feasibility of deploying the proposed security framework in terms of energy efficiency.

Our research evaluates the power overhead through the theoretical analysis and the simulation results. The theoretical analysis is conducted based on a given power consumption model which is built up according to the experimental results of energy consumption behavior of wireless devices [95][96]. The simulation quantitatively evaluates the energy consumption overhead of the proposed security framework in MANETs. It utilizes the simulator GloMoSim 2.02 [17] to implement the access control protocol and the online traceback protocol, respectively, while incorporating the power consumption model. By comparing the energy consumed under the deployment of the security framework with that of the system without security framework, our research shows that the proposed security protocols impose limited amount of power overhead.

1.3 Dissertation Outline

The rest of this dissertation is organized as follows: Chapter 2 provides a background and an overview for the realm of ad hoc networks security, including fundamental cryptographic
schemes, the proposed threat models, and a number of relevant countermeasures. Chapter 3 presents the network access control which prevents an outsider attacker from false data injection. Chapter 4 presents the analyses of online traceback for MANETs and our novel traceback scheme that traces the areas (hotspots) where malicious attacker or compromised intermediate nodes reside. Chapter 5 presents the evaluation of the power overhead of the network access control and the online traceback in ad hoc networks. Finally, Chapter 6 presents the research conclusions and future directions.
Chapter 2

Background

This chapter briefly surveys previous works related to the research topics as outlined in chapter 1. The chapter mostly includes the research in network access control, the traceback, and the power efficiency techniques as reported in the literature.

2.1 Network Access Control

Most ad-hoc networks do not have any provisions for restricting the traffic that flows through a node, i.e., they do not implement any network access control [43]. This leaves these networks vulnerable to packet injection attacks where a malicious node injects a large number of packets into the network with the goal of depleting the resources of the nodes relaying the packets. The packet injection attack must be addressed for the successful deployment of ad-hoc networks due to the constrained resources of mobile nodes.

A packet injection attack is even more effective if an injected packet by a malicious node ends up being multicast or broadcast throughout the network. For example, the operation of most routing protocols involves steps in which a control packet, e.g., a route request packet, is broadcast to all nodes. Moreover, many applications for ad-hoc networks are group-oriented and involve collaborative computing [1]; thus multicast communication is likely to gain more popularity as multicast routing protocols for ad-hoc networks become more mature. Compared to the channel jamming attack, which only affects a relatively small area around the malicious node and could be addressed by techniques such as spread spectrum, channel surfing, or spatial retreat
[38], the packet injection attack using broadcasting or multicasting messages may be more favorable to an attacker due to its network-wide harm.

Clearly, a network access control capability is essential for preventing packet injection attacks in an adversarial environment. In majority of routing protocols for ad-hoc networks, a node that trusts its neighbors will forward packets and assumes that the received packets from its neighbors are authenticated [23][32]. This naive trust model allows a malicious node to inject erroneous routing requests or routing updates into a network, which can paralyze the entire network. For example, Figure 2-1 [16] illustrates an attack that the adversary $M$ changes the network routing paths to a loop by spoofing of route reply (RREP) packets. As shown in Figure 2-1(a), the current network topology among the five nodes $A$, $B$, $C$, $D$, and $E$ allows nodes $A$ and $B$ send data packets to a remote destination $X$. In the network, $A$ can hear $B$ and $D$; $B$ can hear $A$ and $C$; $D$ can hear $A$ and $C$; $C$ can hear $B$, $D$, and, $E$; $E$ can hear $C$ and the next hop on the path toward $X$. Assume a malicious node $M$, which moves around the network, can hear $A$, $B$, $C$, and, $D$. $M$ can learn the network topology by overhearing the route request (RREQ)/route reply (RREP) messages of the network during the phase of route discovery in DSR or AODV protocols. To launch an attack, $M$ impersonates $A$ by changing its MAC address or IP address to match that of $A$, and moves to the vicinity of $B$ which is out of $A$’s radio range. It then sends a RREP to $B$ which contains the hop count to $X$ less than the one sent by $C$ previously. $B$ determines that the better route to $X$ is the path going through $A$, and then changes its route from $C$ to $A$ as illustrated in Figure 2-1(b). $M$ then moves closer to $C$, impersonates $B$, and sends another RREP which makes $C$ believes that the better route toward $X$ is the route going through $B$. Figure 2-1(c) depicts that $M$ finally forms a loop and $X$ is unreachable from the four nodes $A$, $B$, $C$, and $D$. 
To deal with such attacks, recently several security extensions [16][19][20][33][39], have been proposed for authenticating the routing control packets in the network. Dahill et al. [16] identified several security vulnerabilities in AODV [32] and DSR [23], and proposed application of asymmetric cryptography for securing ad-hoc routing protocols. Yi, Naldurg, and Kravets [39] presented a security-aware routing protocol which uses security (e.g., trust level in a trust hierarchy) as the metric for route discovery between pairs. Hu, Perrig, and Johnson designed SEAD [19] for securing DSDV and Ariadne [20] for securing DSR. For instance, Figure 2-2 [2] illustrates the secure routing message exchange in ARAN [16]. The notations used in the figure are as follows:

- \([M]_{Kx}:\) The message \(M\) is signed by the node \(X\) with its private key \(Kx\).
- \(\text{CERT}_X:\) The certificate of the public keys of the node \(X\)
- \(\text{REQ}:\) Route request message.
- \(\text{REP}:\) Route reply message.
- \(S/D:\) The identities of the source and the destination.

ARAN uses public cryptography to ensure that every node on the path knows the correct next hop. As shown in Figure 2-2 [2], the source \(S\) requests a route toward the destination \(D\) by flooding a signed RREQ message. Suppose \(A\) is the next hop, it sets up state of a reverse path pointing to \(S\) upon receiving the RREQ. \(A\) then signs the RREQ, attaches its certificate to prove the authenticity of its public keys, and broadcasts the RREQ. Finally, the RREQ will reach the destination \(D\). When \(D\) receives the RREQ, it generates a RREP, signs it, and unicasts it backward to \(S\) along the reverse path. Upon receiving the RREP, every node along the reverse path verifies the signature and forwards the RREP to the next hop. This process continues until the RREP reaches the source \(S\).
path signs the RREP, attaches its certificate, and relays the RREP to the next hop. For example, upon receiving the RREP from B, A verifies the signature of B using the B’s public key included in CERTb. It then signs the RREP with its private key Ka-, attaches its certificate CERTa to the RREP, and relays the RREP to S. After the secure routing message exchange, a path between S and D is constructed; each node on the path knows the correct next hop. An attacker cannot forge a RREP message without knowing the sender’s private key.

![Figure 2-2: The sequence of secure routing message exchange in ARAN.](image)

We note, however, that none of the proposed secure routing protocols include any provisions for authenticating data packets although data packets are the main traffic in an ad-hoc network. The simplest approach to provide network access control is to employ a network-wide key shared by all nodes. Every node uses this shared key to compute message authentication codes (MACs) on the packets it sends and verify packets from its neighbors. Despite its simplicity, this scheme has several disadvantages. First, an attacker only needs to compromise one node to break the security of the system. Second, if the global key is divulged, it is difficult to identify the compromised node. A compromised node may launch various attacks impersonating other nodes due to the lack of source authentication. Third, it is expensive to recover from a compromise because it usually involves a group key update process. In practice, a system administrator might have to manually reset the group key in the configuration of every user’s wireless network interface card (NIC).
Instead of using a network-wide key, one may use pairwise keys for authenticating every packet. However, when a node broadcasts a packet, it has to attach n MACs to the packet, where n is the number of its immediate neighbors. Consequently, this approach becomes very inefficient for networks with high node density. Source node signing every packet based on public key cryptography can provide network access control; however, due to its large overhead, it is even prohibited in wired networks. An efficient network access control protocol in MANETs should only apply lightweight cryptographic techniques such as symmetric encryptions or hash functions.

In [43], Zhu et al. proposed a Lightweight Hop-by-Hop Authentication Protocol for MANETs (LHAP), a network access control protocol for preventing resource consumption attacks in ad-hoc networks. LHAP uses a one-way key chain (see section 2.1.1) to authenticate data packets; it uses Timed Efficient Stream Loss-tolerant Authentication (TESLA) [46] keys to maintain the trust with neighbor nodes and update the one-way key chain by sending the commitment of the new key chain authenticated by a TESLA key. Since the traffic authentication and the trust management only incorporates hash computations, the protocol is efficient and can be applied to the entire traffic. Thus, an outsider will not be able to inject a packet into the ad-hoc network; the DoS attack will be constrained locally. Figure 2-3 shows that node A joins the network by broadcasting a JOIN message which includes its key commitments authenticated by its signature.
This section introduces the Timed Efficient Stream Loss-tolerant Authentication (TESLA) [46]. TESLA is a broadcast or multicast authentication protocol applying a variant of the one-way key chain technique.

**Definition 2.1: Hash Key Chain**

Suppose $H$ is a one-way hash function, a one-way key chain of length $N+1$ is an ordered set of keys $(K(0), K(1), K(2), \ldots, K(N))$ where $K(N)$ is a random number. $K(N-1) = H(K(N))$, $K(N-2) = H(K(N-1)), \ldots, K(0) = H(K(1))$ where $K(0)$ is referred to as the commitment of this key chain. ■

Figure 2-4 schematically illustrates the TESLA key chain $(K(0), K(1), K(2), \ldots, K(N))$ and the derived key chain $(K'(0), K'(1), K'(2), \ldots, K'(N))$ where $K'(i) = F'(K(i))$, $F'$ is a one-way function. Top of the Figure 2-4 depicts the key generating direction; notice that time advances in a reverse direction. TESLA divides time into intervals of uniform duration. For each of the interval $i$, MAC key $K'(i)$ is used to authenticate the packets sending during that period of time.

A packet sender generates the entire key chain in advance before the first packet is sent out. At

---

Figure 2-3: A scenario in LHAP when node A joins the network. B, C, D, E are its current neighbors. The graph is the current network topology.
the bottom of the figure, we can see the distribution of the packets that the sender sends in each time interval. For example, packets $P_{j+1}$ and $P_{j+2}$ are sent in interval $i$. The sender computes the $MAC$s of the packets $P_{j+1}$ and $P_{j+2}$ using key $K'_i$.

Previous paragraph explains how to generate a TESLA key chain, and how a sender authenticates every packet using $MAC$ keys; this paragraph explains how a sender discloses TESLA keys to recipients and how a recipient authenticates the packets. In TESLA, the sender appends a $MAC$ to every packet, using the key corresponding to the current time interval. The key remains secret for one or several time intervals known as key disclosure delay, and then discloses the key to recipients by embedding the key in the packets dispatched during the delayed time interval.

Figure 2-4: TESLA key chain.
As shown in Figure 2-5 [46], the sender authenticates the packet $P_i$ using key $K'_i$. Suppose the key disclosure delay is one time interval; $K'_i$ is attached to the packet $P_{i+1}$ sending in the next time interval. When the recipient obtains the packet $P_i$, it temporarily stores $P_i$ in a buffer, and updates $K_{i-1}$ as the latest key disclosed by the sender. When the recipient receives the packet $P_{i+1}$ later, it first verifies the key by checking $K_{i-1} = H(K'_i)$. Then it uses $K'_i$ (computing $K'_i = F'(K'_i)$) to authenticate the packet $P_i$ by computing the MAC of $P_i$ with $K'_i$.

### 2.1.2 A Lightweight Inter-layer Protocol (LIP)

The use of TESLA in LHAP leads to some inherent difficulties. First, TESLA requires periodic key disclosure, thus introducing some constant bandwidth overhead that is independent of the actual traffic rate. Second, since TESLA introduces delayed packet verification to forwarding nodes, the use of TESLA makes LHAP vulnerable to an outsider attack for up to one TESLA period.
In this proposal, we propose a novel *Lightweight Inter-layer Protocol* (LIP) to provide an immediate packet authentication. It can prevent outsider attacks and thwart specific insider attacks as well. Moreover, LIP aims to incur smaller bandwidth overhead than LHAP does.

It should be noted that based on threshold cryptography, Zhou and Haas [41] and Luo et al. [25] have proposed hierarchical and distributed network access control schemes for ad-hoc networks. However, the focus of these works is on membership management regarding node join authorization and node revocation. Since these schemes and LIP address different issues, they can be employed in parallel.

Table 2-1: Comparative analysis of access control schemes.

<table>
<thead>
<tr>
<th>Goal</th>
<th>Communication overhead</th>
<th>Cryptographic primitive</th>
<th>Clock synchronization</th>
</tr>
</thead>
<tbody>
<tr>
<td>SEAD [19]</td>
<td>Authenticate routing control</td>
<td>Light</td>
<td>TESLA key chain</td>
</tr>
<tr>
<td>Ariadne [20]</td>
<td>Authenticate routing protocol</td>
<td>Light</td>
<td>TESLA key chain</td>
</tr>
<tr>
<td>AODV-S [115]</td>
<td>Authenticate routing protocol</td>
<td>Heavy</td>
<td>Public key</td>
</tr>
<tr>
<td>LHAP [43]</td>
<td>Network access control</td>
<td>Light</td>
<td>TESLA key chain</td>
</tr>
</tbody>
</table>
2.2 Traceback

The Mobile Ad-hoc network (MANET) paradigm does not rely upon the infrastructure support, however, traditionally it has adopted many existing network protocols such as TCP/IP and hence, it inherits the vulnerabilities of wired networks and suffers from similar attacks (e.g. IP addresses spoofing attacks). In Internet, one applies an Internet Protocol traceback (IP traceback) scheme to trace the attack source when the attacker spoofs the source address of the packets. Traceback in MANETs are more challenging than IP traceback due to the unreliability of relaying nodes and the dynamic topology caused by mobility. It should be noted that some of the IP traceback techniques are still applicable to MANETs traceback (e.g. the packets marking and packets logging schemes). In this section, we review related work both in the realm of IP traceback and the recently proposed traceback schemes for MANETs.

2.2.1 IP Traceback

Figure 2-6: Controlled Flooding.

- **Controlled Flooding**: At the beginning, IP traceback has to be done manually through the cooperation among Internet Service Providers (ISPs). The procedure was tedious and
economically less incentive to many ISPs. To overcome this deficiency, the automatic traceback scheme [53], also known as Controlled flooding, was proposed by sending extra load (UDP packets) to probe each upstream link based on a given victim-rooted route map. The scheme relied on the fact that during DoS attacks the links of the attack path should be heavily loaded. Adding extra load to the suspected links should cause the links become overloaded and a drop in the attack packets rates should be observed by measuring incoming traffic to the attacked system. As shown in Figure 2-6, the process starts from the victim node, and repeats searching for the next hop along the attack path until the attack source is identified. This scheme can only trace uni-source DoS attack and requires ISPs to provide some services. Furthermore, frequently probing is annoying to routers and deteriorates the network performance under Distributed Denial of Service (DDoS) attack.

- **ICMP traceback:** Figure 2-7 illustrates the ICMP traceback scheme [51]. Every network router statistically picks up a forwarding packet (e.g. 1 out of 20,000 packets) and generates an ICMP (iTrace) packet directed to the same destination as the selected packet. The iTrace packet records the previous and next hops, and accommodates as many bytes as possible from the picked packet to its payload. The time to live (TTL) field of the iTrace packet is set to 255 so that the victim node can judge the distance from the router that generates the iTrace packet according to the TTL. If the victim is under a DoS attack, the amount of traffic going to the victim is assumed to be large. Then the victim will eventually get all of the iTrace packets from the routers that implement iTrace on the attack path. By sorting these router addresses in order of TTL, the victim can reconstruct the attack path. The standardization of the ICMP traceback mechanisms, initiated by IETF working group [51], has resulted in an improved scheme [59] that makes iTrace message more useful. Nevertheless, many DDoS attacks also apply ICMP packets to launch an attack. As a result, the application of ICMP traceback is
limited because the defense system (e.g. firewalls) will constrain the ICMP traffic in order to prevent the ICMP flooding attack.

![Diagram of ICMP Traceback](image)

**Figure 2-7: ICMP Traceback.**

- **Probabilistic Packet Marking (PPM):** The probabilistic packet marking (PPM) schemes take a different approach in collecting the intermediate routers information. Instead of generating extra messages which will burden the original traffic load, the routers are configured to randomly mark the packets that pass through them with their addresses or part of their addresses. Each router decides to mark the relaying packets or to update the distance field of the marking which was marked by an upstream router at a fixed probability. The marking done by a downstream router overwrites the marking done by an upstream node. A victim can reconstruct the attack path by gathering the marked packets and applying a path reconstruction algorithm, even though the attacker spoofs the source IP address. Figure 2-8 briefly depicts this scheme. The Attacker launches an attack to the Victim. Suppose that the attack path is R1-R2-R7-R8-R10. R7 is one of the PPM-enabled routers on the attack path. It marks the forwarding packets with a fixed probability $p$ ([64] suggests $p=0.04$). After receiving enough marked packets, the Victim can recover the addresses of all the PPM-
enabled routers along the attack path. Savage et al., [64] introduced the original packet marking scheme which marks a packet in the 16 bits ID field of an IP packet header. The drawback of this scheme is the imposed computational overhead due to the fact that the scheme divides one marking (edge-id) of 72 bits into several fragments of 16 bits to fit the marking into a set of packet headers. As a result, the victim has to compute all the combinations of these fragments to recover the original edge-id. In addition, Song and Perrig [67] have shown that this approach also suffers from false positive rate when the DDoS attack is launched by multiple attackers ($\geq 25$). The improved scheme relies on the assumption that victim knows the map of upstream routers. Thus there is no need to recover the original IPs for all the nodes along the path. Instead, the attack paths are reconstructed by bread-first search of the router-map of victim comparing each node to the information gathered from packet markings.

**Deterministic Packet Marking (DPM):** The probabilistic IP traceback schemes aim to detect the DDoS attack comprised of a large amount of packets (flooding). A deterministic IP traceback, on the other hand, focuses on handling attacks composed of a small number of packets. Belenky et al. [50] proposed a DPM by marking all the ingress packets at the ingress edge routers of the network. This scheme requires the edge router closest to the source to mark its IP address to the ID field of all the incoming packets, and keep the mark unchanged until the packets reach the victim. The scheme is simple, efficient, and only edge routers engaged in packet marking. However, it requires all of the routers implement the protocol (non-incremental) and vulnerable to compromised routers.
Packet Logging: Snoeren et al., [65] developed a packet logging scheme called Source Path Isolation Engine (SPIE), also known as the Hash-based traceback. Unlike packet marking mechanisms which actively modify the content of packets, SPIE passively audits the traffic flow and records packet digests (hash values) for a later forensic analysis. The packet digest is generated by hashing the packet invariant parts. In contrary to the previous schemes, the hash-based traceback has a prominent feature that it can traceback single packet. Therefore, it can be apply to not only DDoS attacks but also the attacks with small amount of packets. Figure 2-9 schematically describes the SPIE architecture. In this scheme, every router, also called data generation agents (DGAs), records partial information of every packet passing through it for the future investigation. In Figure 2-9, R2 is a DGA which hashes every packet passing through it and stores the hashed values into a space efficient data structure called Bloom filter. SPIE divides the network into logical regions. For each of the region, there is a SPIE collection and reduction agent (SCAR) which is responsible to query the traceback information from all the DGAs in that region. A central control unit called SPIE traceback manager (STM) communicates to the victims and SCARs.
The traceback process is conducted in the following steps:

Step1: A victim initiates a traceback request and sends the attack packet digest to the STM.

Step2: The STM sends the requests to SCARs.

Step3: SCARs request the digest tables from DGAs for the appropriate time period.

Step4: After analyzing and correlating the tables received from DGAs, a SCAR can find out which DGAs in its region forwarded the requested packet. Thus, it can reconstruct a part of the attack path if the attack packet passed through its region.

Step5: SCARs report query results to the STM. Based on this information, the STM can reconstruct the attack path throughout the network.

The drawbacks of SPIE are: 1) the communications among SPIE system components are vulnerable to network congestion which may cause the traceback operations fail to complete. 2) The Bloom filter introduces a false positive rate. (i.e. DGA could report a packet passed through it, but actually the packet did not pass through it before). 3) The scheme requires large storage space to accommodate the digest tables. Lee et al. [68] further reduced the memory requirement by hashing aggregated packets instead of individual packets.
granulate individual packets into larger packet aggregating units: flow and source-destination set.

- **Hybrid Scheme:** Gong and Sarac [75], proposed a hybrid IP traceback scheme which adopts both packet marking and logging approaches in an attempt to achieve lower storage overhead and single packet traceback at the same time.

We summarize the traceback schemes in Table 2-2.

Table 2-2: Comparative analysis of IP traceback schemes.

<table>
<thead>
<tr>
<th>IP Traceback Schemes</th>
<th>Scalability</th>
<th>Knowledge of Network Topology</th>
<th>Required Packet Samples</th>
<th>Router Assistance</th>
<th>Vulnerable to Compromised Routers</th>
</tr>
</thead>
<tbody>
<tr>
<td>iTrace [51] (ICMP)</td>
<td>Poor</td>
<td>Non-required</td>
<td>Large</td>
<td>Non-required</td>
<td>Yes</td>
</tr>
<tr>
<td>FMS [37] (PPM)</td>
<td>Poor</td>
<td>Non-required</td>
<td>Large</td>
<td>Non-required</td>
<td>Yes</td>
</tr>
<tr>
<td>AMS [36] (PPM)</td>
<td>Good</td>
<td>Required</td>
<td>Median</td>
<td>Non-required</td>
<td>No</td>
</tr>
<tr>
<td>DPM [50]</td>
<td>Good</td>
<td>Required</td>
<td>Median</td>
<td>Required</td>
<td>Yes</td>
</tr>
<tr>
<td>SPIE [65] (Logging)</td>
<td>Good</td>
<td>Required</td>
<td>Single</td>
<td>Required</td>
<td>No</td>
</tr>
</tbody>
</table>
2.2.2 Traceback in MANETs

At present, within the scope of the MANETs infrastructure, only a few traceback schemes have been advanced in the literature. Huang and Lee [58] developed a hotspot-based traceback protocol, in which every intermediate node records the neighbor list and the time-to-live (TTL) value of each forwarding packet in a Tagged Bloom Filter (TBF), which not only logs the packet digest of the forwarding packets but also logs the TTL value in their headers. In the traceback request phase, an investigator broadcasts a query that contains the digest of the attack packet to the entire network and then collects the reports from all of the nodes that have previously forwarded the packet. After gathering the reports from all of the matched nodes, the investigator obtains a snapshot of the network topology and then runs hotspot detection algorithms to identify single or multiple approximate locations (hotspots) where the adversaries reside. The broadcast nature of this protocol implies high communication overhead that would incur flooding-based DoS attacks.

Kim and Helmy proposed a DoS attacker traceback scheme called SWAT [73]. In SWAT, every node builds traffic profile consisting of traffic pattern and traffic volume. When a victim detects a DoS attack, it initiates a traceback request containing the abnormal traffic profile to search for the attack path. To facilitate this search, each node maintains information about its vicinity nodes (within $R$ hops away) and its contact outside the vicinity ($R+r$ hops away). The goal is to reduce the communication overhead of flooding by forwarding the request to a number of vicinities rather than the entire network. This work, however, has several shortcomings: First, although building traffic pattern potentially incurs less storage overhead than a logging scheme, it could result in a high false positive rate during profile matching due to the statistical nature of network traffic, especially for low-rate attacks. Second, the constant overhead of maintaining ($R+r$)-hop topology information for the purpose of possible traceback is prohibiting, while both
of the broadcast-based scheme [58] and ours are on-demand and there is no need to maintain
 topology information.

Kim and Helmy also accomplished a more detailed analysis for the traceback scheme
based on traffic analysis in ATTENTION [74]. This work basically uses the same traffic
abnormality detection paradigm of SWAT; in addition, they further consider the mobile DoS
attacker and the traffic information fusion. They use the spatial relation and temporal relation to
distinguish the mobile DoS attack from the distributed DoS attack. The spatial relation is that a
specific traffic pattern (the attack signature) which is observed on various locations; to a mobile
attacker, these observed locations are supposed to be continuously (spatial continuity). The
temporal relation is the various time slots in which the attack signature is observed. If an attack
signature is observed in the same time slot but at different locations, which implies a distributed
DoS attack. Their experimental results indicate that ATTENTION has generally higher traceback
success rate, pxc24@ntnu.edu.tw however it also possesses a higher false positive rate. In the
scenario of distributed DoS attack, their simulation results show that ATTENTION is more robust
than SWAT; as the number of attacker increased the traceback success rate of SWAT decreases
more quickly than that of ATTENTION. The reason is that when the number of attack source
increased, the abnormal characteristic of attack traffic is concealed in the normal background
traffic.

Table 2-3: Comparison of traceback schemes in MANETs.

<table>
<thead>
<tr>
<th>Ad Hoc Traceback Schemes</th>
<th>Traceback Approach</th>
<th>Communication Overhead</th>
<th>Storage Overhead</th>
<th>Required Packet Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWAT [73]</td>
<td>Cluster-based</td>
<td>Heavy</td>
<td>Light</td>
<td>Large</td>
</tr>
</tbody>
</table>
2.3 Power Efficiency

![Power consumption of a mobile device iPAQ](image)

Figure 2-10: Power consumption of a mobile device iPAQ [105].

As indicated in Figure 2-10 [105], study shows that energy consumed by wireless interface represents the main portion of battery power consumption of a mobile device. The power consumed for the wireless communication constitutes almost half of the overall power consumption of the mobile device. As noted in [114], careful prediction of the upcoming application queries reduces the energy consumption, significantly. As reported in [95], wireless network interface operating in ad hoc mode consumes even more energy than that operating in base station (BSS) mode. Hence, energy conservation is of great concern in ad hoc network protocol design. In this section, we introduce a power consumption model for ad hoc networks,
and a brief survey of the wake-up based power management approaches that commonly used in
the network layer of the ad hoc networks.

2.3.1 Power Consumption Model

Power-aware network protocol design and the energy efficiency evaluation for the ad hoc
network rely on the realistic knowledge of the power consumption behavior of the wireless
interface operating in the ad hoc networking environment. It is useful to model the energy
consumption behavior in a form that is convenient for the protocol design and analysis. In [95], a
series of experiments had been conducted which measured the energy consumption of a Lucent
WaveLAN IEEE 802.11 wireless network interface card operating in the ad hoc mode. The data
obtained by the detailed measurements is presented in the form of linear equations for point-to-
point packet sending/receiving, broadcast sending/receiving, promiscuous receiving/discarding,
and idle state, respectively. This model considers the energy consumption in a per-packet
granularity level which is very useful for evaluating the power overhead of our proposed network
access and traceback protocols (Chapter 3 and 4).

The per-packet energy consumption model of wireless network interface in ad hoc mode
is presented as follows:

\[ \text{Energy} = m \times \text{size} + b. \]  \hspace{1cm} (2-1)

The equation consists of a fixed component \( b \) and an incremental component \( m \times \text{size} \). The
coefficient \( b \) represents the overhead of channel acquirement and device state change, the
coefficient \( m \) is the constant of proportionality, and \( \text{size} \) represents to the amount of data.
Equation (2-1) shows that the energy consumption is proportional to the size of packet being
transmitted, received, or dropped. The model considers different energy consumption behaviors
for each network interface reactions; and hence, different $m$ and $b$ values are assigned to sending, receiving, and discarding packets, and promiscuous and point-to-point transmission.

Table 2-4: Lucent IEEE 802.11 WaveLAN PC CARD (2 Mbps) [95].

<table>
<thead>
<tr>
<th>Mode</th>
<th>$\mu W \cdot \text{sec/byte}$</th>
<th>$\mu W \cdot \text{sec}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Point-to-point send</td>
<td>1.9 $\times$ size +454</td>
<td></td>
</tr>
<tr>
<td>Broadcast send</td>
<td>1.9 $\times$ size +266</td>
<td></td>
</tr>
<tr>
<td>Point-to-point receive</td>
<td>0.50 $\times$ size +356</td>
<td></td>
</tr>
<tr>
<td>Broadcast receive</td>
<td>0.50 $\times$ size +56</td>
<td></td>
</tr>
<tr>
<td>Promiscuous receive</td>
<td>Non-destination host in range of both sender and receiver.</td>
<td>0.39 $\times$ size +140</td>
</tr>
<tr>
<td>discard</td>
<td>-0.61 $\times$ size +70</td>
<td></td>
</tr>
<tr>
<td>Promiscuous receive</td>
<td>Non-destination host in range of sender.</td>
<td>0.54 $\times$ size +66</td>
</tr>
<tr>
<td>discard</td>
<td>-0.58 $\times$ size +24</td>
<td></td>
</tr>
<tr>
<td>Promiscuous receive</td>
<td>Non-destination host in range of receiver.</td>
<td>0.0 $\times$ size +63</td>
</tr>
<tr>
<td>discard</td>
<td>0 $\times$ size +56</td>
<td></td>
</tr>
<tr>
<td>Idle (ad hoc) (k)</td>
<td>843 mW</td>
<td></td>
</tr>
<tr>
<td>Idle (base station)</td>
<td>66 mW</td>
<td></td>
</tr>
</tbody>
</table>

Table 2-4 [95] shows the coefficients of the linear formulas modeling energy cost for different operating modes; the coefficient values are determined based on the experimental results. Notice that the energy cost in this table are defined relative to the power consumption of the wireless
interface under the *idle* mode — hence some slopes (incremental cost) appear negative symbol. From the entries in Table 2-4, one can make a number of observations as follows:

1. The proportional coefficient $m$ of the point-to-point sending and the broadcast sending are the same; however, the fixed overhead $b$ for the point-to-point sending is higher than that of the broadcast sending because the point-to-point traffic uses extra control packet (RTS) to support media reservation while the broadcast traffic does not support that.

2. The fixed overhead for the data receiving for the point-to-point traffic is significantly higher than that of the broadcast traffic because the point-to-point traffic receiver sends two control messages (CTS/ACK) for each traffic.

3. A non-destination host in the range of the sender enters a reduced power consumption mode and ignores the packet being transmitted; the energy consumption at this mode is lower than the idle mode. Therefore the incremental cost of promiscuous receiving is negative.

4. A non-destination host $U$, which resides out of the radio range of the sender $S$, does not overhear the point-to-point data from $S$. The incremental cost of $U$ in Equation (2-1) is equivalent to the cost of idle mode with a slope zero. Additionally, in contrast to the nodes within the transmission range of $S$, $U$ is more likely to receive the broadcast from other senders without colliding with the traffic from $S$; the fixed energy cost of $U$ in Equation (2-1) is equivalent to that of a broadcast receiving node.

The energy consumption model described above will be used in the evaluation of the power overhead of the proposed security framework in this research.
2.3.2 Wake-up based power management

Wireless network interface operating in ad hoc mode persistently overhears the wireless channel due to the lack of base stations to coordinate the communication. Constant idle power consumption is thus imposed to the network nodes when there is no traffic. Research shows that the power consumption in idle mode is only slightly less than that of receiving data traffic and it makes up a significant portion of the overall power consumption of wireless interface [95]. The wake-up based power management intends to save the energy wasted in the idle mode.

Wake-up based power management is a commonly used strategy in network layer of ad hoc networks. It utilizes the 802.11 power save mode (IEEE 802.11 PSM) to reduce the power consumption of idle nodes. Typically, an 802.11 complied wireless network interface operates in one of four different states:

- **Transmit State**: the interface is sending a packet.
- **Receive State**: the interface is receiving a packet.
- **Idle State**: the interface is constantly listening to the wireless channel to determine whether or not there is a packet to receive or to send.
- **Sleep State**: the interface enters a low power consumption mode while the transmission and reception are disabled.

Table 2-5 shows a typical power consumption model used in the evaluation of wake-up based power management protocols.

Table 2-5: Power Consumption Model [99].

<table>
<thead>
<tr>
<th>States</th>
<th>Power</th>
</tr>
</thead>
<tbody>
<tr>
<td>Transmit</td>
<td>1400 mW</td>
</tr>
<tr>
<td>Receive</td>
<td>1000 mW</td>
</tr>
<tr>
<td>Idle</td>
<td>830 mW</td>
</tr>
</tbody>
</table>
Current wake-up based power management protocols can be classified into three categories: the scheduled rendezvous protocols [99][100][101], the asynchronous protocols [97][104], and the on-demand protocols [96][98]. We briefly describe each of them in the following.

- **Scheduled rendezvous**: The scheduled rendezvous protocols require every node periodically wake up at about the same time to check if there are packets cached for receiving/delivery. The SPAN protocol operates between the network layer and the MAC layer [99]. It defines the rules to coordinate the sleeping nodes and the awake nodes in a distributed manner. In SPAN, nodes are partitioned as master nodes and slave nodes. The master nodes are selected to maintain the network connectivity. In other words, the master nodes serve as the network backbone. The master nodes play the role of access point (AP) in IEEE 802.11, they are kept awake all the time and periodically sending a beacon and the Traffic Indication Map (TIM) to wake up the slave nodes as packets are needed to be delivered to them. In [101], Xu et al. proposed GAF in which the network nodes are assumed knowing their geographic locations. Similar to SPAN, a subset of nodes are selected as delegates serving for packet routing, periodically sending out a beacon to the sleeping nodes. GAF partitions territory into grids and nodes are associated to the virtual grids (the grid size is related to the radio transmission range of the network nodes). For each virtual grid, one node is selected as the delegate serving that geographic area. Both of GAF and SPAN ensure the connectivity of the network by keeping a small subset of the nodes in active operational mode for delivery packets while the rest of the nodes are asleep. When the nodes are mobile, constant communication cost is incurred in order to determine the role of each node.
• **Asynchronous**: The asynchronous scheduling relieves the requirement of the clock synchronization among the network nodes. By carefully designing a wake-up schedule, the asynchronous schemes ensure that every two neighboring nodes (within the transmission range) may discover each other within a bounded time no matter how their clock drifts from each other. More specifically, the asynchronous scheduling scheme guarantees an overlap of any two wake-up schedule in a cycle. For example, consider Figure 2-11. Node A and node B have the same wake-up schedule pattern with block shift. Node A can detect the beacon from node B and node B can listen to the beacon from node A somewhere in a repetitious schedule cycle.

![Figure 2-11](image.png)

Figure 2-11: An example that shows two neighboring node can listen to each other with clock drifts [97].

Tseng et al. [104] proposed to use the concept of quorum in designing the wake-up schedule. The schedule cycle is arranged in an \( n \) by \( n \) square; each cell represents a beacon interval and is arranged in the order of the index. Every node chooses a row and a column from the square as the wake-up schedule. Since a column crosses any other row in the square, the scheme guarantees that every two nodes can detect each other within the schedule cycle. Figure 2-13, for example, shows two wake-up schedules. The left schedule and the right schedule overlap on two intervals no.2 and no.9.
Zheng et al. [97] proposed a systematical way to design the wake-up schedule with the *wakeup schedule function* (WSF). The problem of asynchronous scheduling was transformed to the problem of block design in the theory of combinatorics, which resulted in a theoretical foundation for optimizing the solution that minimizes the idle state with bounded neighbor discovery latency. Figure 2-13 illustrates a schedule designed using the (7,3,1) pattern, where each schedule cycle consists of 7 slots (beacon intervals), 3 out of 7 slots are awake slots, and it is guaranteed that each pair of schedules overlap on 1 slot; the figure shows all the possible schedule shifts with the slot boundary aligned (the condition of slot boundary alignment can be released). A node wakes up in three slots out of a seven-slot cycle. No matter how the clock is shifted, any two nodes will detect each other at exactly one slot (any two rows in this figures overlap at one slot).

![Figure 2-12: Quorum based wake-up schedule. The filled cells indicate the beacon intervals at which the node wakes up. [104].](image)

![Figure 2-13: A (7,3,1) WSF design where 3 slots are ON slots out of a 7-slot cycle. [97].](image)
According to the analysis in [105], the asynchronous scheme possesses a number of advantages over the scheduled rendezvous: First, this scheme requires very little coordination among the network nodes which implies the diminished communication as well as the energy overhead. Second, each node chooses basically the same wake-up schedule. Hence, the power consumption among the network nodes is balanced which by default extends the network lifetime. Finally, unlike the synchronous schemes (i.e., scheduled rendezvous), the asynchronous scheme is not subject to the network density. Thus, it may work in sparse networks.

The asynchronous approach also has several deficiencies over the synchronous approach: First, the guarantee of contacting any of the neighbors within a bounded amount of time comes at the expense of introducing a generally large amount of hop-by-hop packet forwarding delay in a multi-hop ad hoc network. Second, the asynchronous protocols heavily rely on detecting the beacon signals from neighbors. Therefore, in a dense network, these beacon messages may impact the protocol performance and the probability of message collision. Third, the asynchronous protocol does not allow the neighboring nodes being in the wake up mode at the same time. As a result, it becomes difficult to broadcast a message. For example, consider Figure 2-13. A node must broadcast a message three times to ensure that all of its neighbors being notified. Finally, the asynchronous protocols generally require a higher duty cycle than that of the synchronous ones in order to make sure that a node can discover all of its neighbors. Hence, the asynchronous protocols can barely achieve the same level of power conservation as their synchronous counterparts can achieve.

• **On-demand:** The shortcoming of both scheduled rendezvous and the asynchronous wake-up approach is the difficulty of predicting when the communication is needed between network nodes. To tackle this problem, the on-demand protocols use out-band signal to awaken the sleeping node only when it is needed. Every node is equipped with a secondary wireless
interface which remains ON all the time to listen to a wake-up signal. The secondary wireless interface is supposed to be power efficient and used only for the wake-up purpose. The advantage of using on-demand approach over the aforementioned schemes is that it can pipeline the data forwarding. By concurrently using two wireless interfaces, a relaying node can receive data from the previous hop while notifying the next hop to be prepared to receive data at the same time. As a result, the hop-by-hop data forwarding delay for the on-demand approach is largely reduced in contrast to the previously described wake-up scheduling schemes.

Numerous on-demand wake-up schemes have been proposed in the literature. Nosovic and Todd [106] proposed to apply radio frequency identifier (RFID) technology on wakening up embedded systems. RFID is a simple, mature, and low cost technique; it only consumes energy at around three orders of magnitude lower than those of commercial radios operating in the Mbps range. Unfortunately, the radio range of RFID is also much shorter so that the radio range of wake-up interface and the data communication interface are highly asymmetric; it makes the RFID technique not fit to the scenario of multi-hop ad hoc networks [97]. In [109], the authors used a binary wake-up channel to improve the energy efficiency of the wake-up channel. They integrate the separate channel mechanism with a wake-up schedule which is made based on the observation of the past communication pattern. However, this scheme indiscriminately wakes up all neighbors which results in the power consumption overhead.

The main problem with the on-demand approach is that the secondary radio interface generally does not possess a transmission range that analogous to the primary interface. Consequently, the network nodes operate closer to each other than that would be necessary, and hence, waste of the resource. Moreover, although secondary wireless interface consumes less power than the primary radio (e.g., IEEE 802.11), the current proposed secondary radios
are still not considered ultra low power consumption. For further discussion on on-demand approach, the reader is referred to the [106][107][108][109].

Table 2-6: Comparative analysis of different power saving schemes.

<table>
<thead>
<tr>
<th>Power Efficient Schemes</th>
<th>Clock Synchronization</th>
<th>Network Delay</th>
<th>Additional Radio Interface</th>
<th>Network Topology Management</th>
</tr>
</thead>
<tbody>
<tr>
<td>SPAN [99] (Synchronous)</td>
<td>Required</td>
<td>Median</td>
<td>No</td>
<td>Required</td>
</tr>
<tr>
<td>GAF [101] (Synchronous)</td>
<td>Required</td>
<td>Median</td>
<td>No</td>
<td>Required</td>
</tr>
<tr>
<td>Quorum Based [104]</td>
<td>Non-required</td>
<td>Large</td>
<td>No</td>
<td>Non-required</td>
</tr>
<tr>
<td>(Asynchronous)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WSF [97] (Asynchronous)</td>
<td>Non-required</td>
<td>Large</td>
<td>No</td>
<td>Non-required</td>
</tr>
<tr>
<td>(Asynchronous)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>STEM [108] (On-demand)</td>
<td>Non-required</td>
<td>Small</td>
<td>Yes</td>
<td>Non-required</td>
</tr>
<tr>
<td>(On-demand)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2.4 Conclusion

In contrast to Internet, MANETs encounter more severe challenges from both the security threats and the resource, especially the energy, constraint. Therefore, in the passing few years, the security mechanisms and the power conservation schemes for MANETs have being extensively
studied in the research community. Our research is among the efforts to address these challenges; we proposed our approaches for the unsolved problem and proposed a better solution to improve the performance of the existing work.

In this Chapter, we presented a background knowledge and a survey for our research that include: MANETs routing security, hash key chain technique, hop by hop authentication, IP traceback, online traceback for MANETs, the power consumption model of wireless interface, and the wake-up based power management schemes. Those are the state of the art in the literature for addressing the challenges of security and power efficiency, such as network access authentication in MANETs, insider attack traceback, and power consumption evaluation. Our research either is developed based on these foundations or utilizes them as a tool to evaluate the proposed schemes.

In the next Chapter, we will present the proposed network access control protocol which is a lightweight hop by hop authentication scheme using the hash key chain technique.
Chapter 3

Network Access Control

3.1 Introduction

Most ad hoc networks do not have any provisions for restricting the traffic that flows through a node, i.e., they do not implement any network access control. This leaves these networks vulnerable to packet injection attacks where a malicious node injects a large number of packets into the network with the goal of depleting the resources of the nodes relaying the packets. The packet injection attack must be addressed for the successful deployment of ad hoc networks due to the constrained resources of mobile nodes.

A packet injection attack can be especially effective if a packet injected into an ad hoc network by a malicious node ends up being multicast or broadcast throughout the network. For example, most routing protocols involve steps in which a control packet, e.g., a route request packet, is broadcast to all nodes. Moreover, many applications for ad hoc networks are group-oriented and involve collaborative computing; thus multicast communication is likely to increase in importance as multicast routing protocols for ad hoc networks become more mature. Compared to the channel jamming attack, which only affects a small area around the malicious node and could be addressed by techniques such as spread spectrum, channel surfing, or spatial retreat [38], the packet injection attack using broadcast messages may be more favorable to an attacker due to its network-wide harm.

Clearly, a network access control capability is essential for preventing packet injection attacks in an adversarial environment such as a battlefield. Most of the proposed routing protocols for ad hoc networks do not address the issue of network access control. In these protocols, a node
assumes that its neighbors are trustworthy and hence either they will forward its packets, or it receives authenticated packets from its neighbors. This naive trust model allows a malicious node to inject erroneous routing requests or routing updates into a network, which can paralyze the entire network. To deal with such attacks, recently several security extensions protocols to authenticate the routing control packets in the network have been proposed [16][19][20][33][39]. We note, however, that none of the proposed secure routing protocols include any provisions for authenticating data packets although data packets are the main traffic in an ad hoc network.

The simplest approach to provide network access control is to employ a network-wide key shared by all nodes. Every node uses the shared key to compute message authentication codes (MACs) on the packets it sends and verify packets it receives. Despite its simplicity, this scheme has several disadvantages: First, an attacker only needs to compromise one node to break the security of the system. Second, if the global key is divulged, it is difficult to identify the compromised node. A compromised node may launch various attacks impersonating other nodes due to the lack of source authentication. Third, it is expensive to recover from a compromised network because it usually involves a group key update process. In practice, a system administrator might have to manually reset the group key in the configuration of every user’s wireless NIC.

Instead of using a network-wide key, one may use pair-wise keys for authenticating every packet. However, when a node broadcasts a packet, it has to attach \( n \) MACs to the packet, where \( n \) is the number of its immediate neighbors. Consequently, this approach becomes very inefficient for networks with high node density. As another potential solution, source node signs every packet based on public key cryptography to provide network access control; however, this incurs large overhead which has prohibited its application in wired networks, not mentioning resource scares ad hoc networks.
The rest of this chapter is organized as follows: The protocol assumptions and its design goal are presented in Section 3.2 and overviewed in Section 3.3. Two protocols are introduced in Sections 3.4 and 3.5, respectively. Section 3.6 discusses further deployment issues. Security analysis of the proposed protocol is discussed in Section 3.7. In Section 3.8 and 3.9, we present the experimental results and the performance analysis of our protocol. Finally, Section 3.10 concludes this chapter.

### 3.2 Assumptions and Design Goal

#### 3.2.1 Security Assumptions

We assume that every pair of mobile nodes can establish a pair-wise key on the fly based on an appropriate id-based scheme, e.g., preloading pair-wise keys or probabilistic-polynomials [26], or using standard public key cryptography (if the computational resources of nodes are less constrained). The id-based scheme allows two nodes knowing each other’s id to establish a pair-wise key on-the-fly without requiring the existence of an on-line key server. Moreover, the id-based scheme prevents a node from impersonating another node because it does not possess the keys for that node. We note that secure routing protocols introduced in [16][20] also assume the similar way to bootstrap trust between nodes. Hence, when employing LIP together with secure routing protocols, we do not need to add another mechanism for establishing pair-wise keys between nodes.

We do not address attacks against the physical layer and the media access control layer. Techniques such as spread spectrum, frequency hopping, and spatial retreat [38] can be employed to prevent physical jamming attacks if necessary. Cardenas et al. [14] have studied techniques for detecting and preventing media access control layer attacks.
3.2.2 Security Assumptions

We mainly consider the packet injection attack in which an attacker injects a large number of junk packets into an ad hoc network with the goal of depleting the resources of the nodes that relay the packets. In addition, these packets could introduce severe wireless channel contention and network congestion. The packets could be unicast packets, local (one-hop) broadcast packets, or network-wide broadcast packets. Clearly, the attack is the most effective if the injected packets end up being flooded in the entire network.

The attacker could be an outsider (unauthorized) node that does not possess a valid credential, or an insider (authorized) node that possesses a valid credential. An insider node launches the impersonation attack because it has been compromised or it intentionally does it; we do not distinguish the attack motivation here. We use the term “impersonation” to refer to the case when compromised nodes impersonate non-compromised nodes, not when compromised nodes impersonate each other. To achieve the attack goal, an attacker may eavesdrop, reorder, and drop packets, fabricate packets, replay older packets, or modify overheard packets and re-inject them into network.

An attacker may use its own id, fabricated ids, or spoofed ids as the sources of the injected packets; however, in this work we do not prevent the attack where an insider attacker directly uses its own id. To prevent this type of insider attack, we have to regulate the normal traffic pattern (e.g., the maximum Route Request rate [30]) for each node. The violation of the regulation indicates the compromise of the node and rekeying schemes such as GKMPAN [44] may then be applied to revoke the compromised node.
3.2.3 Design Goal

The goal of this work is to provide an efficient network access control mechanism for preventing packet injection attacks. To achieve this goal, it is essential that a node be able to verify the authenticity of every packet received from other nodes. As a result, the protocol should meet the following requirements:

- **Efficiency:** The protocol must be resource efficient since every packet will need to be authenticated; otherwise, the amount of resources it consumes may be equivalent to that caused by packet injection attacks. Since packet transmission contributes to the main portion of energy expenditure of a wireless node, the protocol should minimize the additional bandwidth overhead.

- **Scalability:** The performance of the protocol, in terms of computational and communication cost, should not degrade with the network size. The scheme should not require every node to have the global knowledge of network.

- **Immediate Authentication:** The protocol should provide immediate authentication, i.e., there should be no latency in authenticating a received packet; otherwise, the latency of packet delivery will be unacceptably high in a multi-hop communication setting and a node might have to dedicate a large memory space for buffering those temporarily unverifiable packets.

- **Transparency:** It is undesirable that the deployment of a protocol requires modification or redesign of other protocols in the protocol stack. Therefore, the protocol should work transparently with other protocols, i.e., the protocol may be turned on or turned off without affecting the functionality of other protocols such as routing protocols or application layer protocols.
• **Independence:** The protocol should work regardless of the deployed routing protocol. It is possible to design a specific and more efficient protocol that works with a specific routing protocol; however, this is not effective given the vast number of routing protocols as advanced in the literature.

### 3.3 Overview

We first present an overview of LIP, then discuss two schemes in detail—a basic scheme, followed by a location-aware version of this scheme.

**Notations:** We use the following notation to describe security protocols and cryptography operations in this chapter:

- $u, v$ (in lower case) are the identities of mobile nodes.
- $M1 \parallel M2$ denotes the concatenation of message $M1$ and $M2$.
- $MAC(K, M)$ denotes the computation of MAC over message $M$ with key $K$.
- $\{M\}_k$ is encrypting message $M$ with key $K$.

The goal of our protocol is to provide full network access control. As such, the protocol does not distinguish between data packets and routing control packets for authentication purposes. For simplicity, we call all these packets as traffic packets. The protocol is transparent to and independent of the network routing protocol. It can be thought of inserting an authentication layer residing between the data link layer and the network layer, providing a protection mechanism that can prevent many attacks from happening. This transparency and independence allows the protocol to be turned on or turned off without affecting the operations of other layers. Figure 3-1 shows the protocol stack.
To minimize packet overhead, we design LIP based on a localized broadcast authentication mechanism in which a node only computes and attaches one message authentication code (MAC) to each traffic packet it is forwarding (or originated from it). For its neighbors to verify its packets, a node must share its MAC keys (referred to as one-time authentication keys hereafter) with its neighbors. Hence, we introduce the one-time authentication key management process for a node to establish and maintain its one-time authentication keys.

The one-time authentication keys of a node should only be used by the node to authenticate its packets to its neighbors while its neighbors use the same one-time authentication keys only for verification purpose. However, due to the symmetry nature of one-time authentication keys, a malicious neighbor may impersonate the node by using the node’s one-time authentication keys to generate MACs over injected packets. To thwart this impersonation attack, we propose three techniques: one-time authentication key, random neighborship verification, and location-aware verification. The use of one-time authentication keys builds the first defense line to prevent the impersonation attack, making the attack very difficult to succeed. The random verification process can further detect such attack in case that sophisticated attacks cross the first defense line. By assuming the availability of node location and velocity information, location-aware verification
verification can further reduce bandwidth overhead. The details of these techniques are presented below.

3.4 Scheme I: Basic Scheme

3.4.1 Using One-time Authentication Keys

The basic scheme uses one-time authentication keys; that is, a node uses an authentication key only once to thwart an attacker from reusing it. One-time authentication keys are provided by the technique of one-way key chains [24]. A one-way key chain is an ordered set of keys generated through repeated application of a one-way hash function $H$ on a random number. For instance, if a node wants to generate a key chain of size $l+1$, it first randomly chooses a key, say $K(l)$, then computes $K(l-1) = H(K(l))$, $K(l-2) = H(K(l-1))$, ..., repeatedly until it obtains $K(0) = H(K(1))$.

Once a node has generated its key chain, it can use the keys in its key chain as one-time authentication keys for authenticating one packet. To enable its neighbors to verify a one-time authentication key in its key chain, a node first bootstraps its key chain by sending the commitment of its key chain, i.e., $K(0)$, to each of its current neighbors, encrypted with their pairwise key. The node then uses a one-time authentication key in its key chain to compute the MAC of a packet it is transmitting. Note that the one-time authentication keys are consumed in an order reverse to that of their generations. A receiver can authenticate $K(j)$ by verifying $K(j-1) = H(K(j))$ if it has $K(j-1)$. Furthermore, if a receiver did not receive $K(j-1)$ and the last key it authenticated is $K(i)$, where $i < j - 1$, it can still authenticate $K(j)$ by verifying $K(i) = H_{j-i}(K(j))$. 
Consider the scenario where node $u$ wants to authenticate a packet $P(i)$ to its neighbors $v_1, v_2, \ldots, v_m$, using $K(i)$ as the MAC key. In the message $M$ that contains $P(i)$, the node embeds its next one-time authentication key $K(i+1)$, and attaches a MAC of $P(i)$ computed with $K(i)$. Assuming that the key chain size is $n$ where $i = 1, 2, \ldots, n$, we present the message $M$ as follows.

$$M : i, P(i), K(i+1) \oplus MAC(K(i),i \parallel P(i))$$ (3-1)

Here we assume the size of a key is the same as the output of a MAC, for example 8 bytes. When a neighbor node $v$ receives the message, it performs three operations. First, it computes $MAC(K(i), P(i))$ based on $K(i)$, which it derives from the previous message. Second, after computing a MAC over $P(i)$ based on $K(i)$, it derives $K(i+1)$ by a bitwise-XOR operation. Finally, it checks if $H(K(i+1)) = K(i)$. If the verification succeeds, it sets $K(i+1)$ as node $u$’s next valid MAC key. In addition, it adds $u$ into its local trust list. Any future packets that are authenticated with a one-time authentication key prior to $K(i+1)$ will be discarded. As a result, an attacker cannot simply reuse the previous one-time authentication keys of node $u$ to deceive a neighbor node $v_j$. Finally, the LIP protocol of node $v_j$ passes the verified packet to the routing protocol for process. If node $v_j$ decides to forward this packet to one or more neighbors, its LIP protocol will use node $v_j$’s one-time authentication key to authenticate the packet to others. As such, a packet is authenticated in a hop-by-hop fashion.

Now we consider the synchronization issue due to unreliable transmission. From Message (3-1) we can observe that if a neighbor $v_j$ of node $u$ lost the previous packet $P(i-1)$ that contains $K(i)$, it will not be able to verify the current packet $P(i)$ or to derive $K(i+1)$ because it does not know $K(i)$. In this case, a simple solution is that the neighbor requests $K(i)$ or $K(i+1)$ from node $u$. A better solution is to use a self-healing key distribution mechanism. Instead of using $K(i)$ for authenticating $P(i)$, we can use an earlier key $K(i-m), m > 0$. If node $v_j$ has $K(i-m)$, it can verify $P(i)$ and derive $K(i+1)$, and then compute all the keys between $K(i-m)$ and
$K(i+1)$ based on the one-way hash function even if it has missed these intermediate keys. Here the choice of $m$ should be determined by the packet loss rate in the network.

The above authentication scheme is motivated by two observations: First, since packets are authenticated hop-by-hop, a node only needs to authenticate a packet to its immediate neighbors. Second, when a node sends a packet, a neighbor will normally receive the packet before it receives a copy forwarded by any other nodes. This is due to the triangle inequality among the distances of the involved nodes, as shown in Fig. (3-2(a). When node $u$ sends a packet that is authenticated with a one-time authentication key $K(i)$ in its key chain, node $v$ normally receives the packet before it receives a forwarded copy from node $x$ because $|uv| < |ux| + |xv|$, unless the packet is lost. Thus, it is difficult for an adversary $x$ to impersonate node $u$ to $v$ by reusing node $u$’s one-time authentication keys.

The one-time key-based scheme however cannot completely prevent the impersonation attack. In Figure (3-2(b), after node $v$ has moved out of the transmission range of node $u$ (but $v$ is unaware of it), it cannot know the most recent one-time authentication key disclosed by $u$. Therefore, node $x$ may reuse $u$’s old one-time authentication keys $K(i)$ and $K(i + 1)$ to inject a false packet $P(i)$ to node $v$ using a directional antenna.

$$M': i, P'(i), K(i + 1) \oplus MAC(K(i), i \parallel P'(i))$$

More complicated attacks may involve more than one colluding nodes. For example, even when node $v$ is still in the range of $u$, another malicious node could manage to jam node $v$ when node $u$ is transmitting. Later node $x$ could send a modified packet to $v$ impersonating $u$ as in the previous case. We note, however, that the number of impersonated packets in this attack is bounded by the actual transmission rate of node $u$, because node $x$ cannot use the one-time authentication keys which node $u$ has not disclosed yet due to the one-way property of a hash function. Moreover, reusing node $u$’s one-time authentication keys within the one-hop range of
node $u$ is subject to detection by $u$ and the other neighbors. Therefore, this scheme provides reasonably strong source authentication and an attacker takes a high risk of being detected when it reuses the one-time authentication keys of other nodes.

3.4.2 Random Neighborship Verification

We now discuss a random neighborship verification scheme to further deter the impersonation attack. The main idea is that a node challenges its neighborship with another node with certain probability. In the example shown in Figure (3-2(b), when node $v$ receives a packet $P(i)$ from a claimed source $u$, it responds with a CHALLENGE message at probability $p_c$:

$$v \xrightarrow{p_c} u : i, MAC(K_{vu}, MAC(K(i), P(i)))$$

(3-3)

where $i$ is the packet index and $K_{vu}$ is the pairwise key shared between $v$ and $u$. To save computational overhead, here node $v$ refers to packet $P(i)$ by $MAC(K(i), P(i))$, which is the $MAC$ contained in $P(i)$ (refer to Message (3-1). Also, node $u$ is required to keep the $MAC$ of every transmitted packet for a short time interval to recognize any challenged packet. If node $u$ can hear this CHALLENGE message, it replies with the following ACK message:
If $\text{FLAG}$ is TRUE, the message proves to $v$ that node $u$ has really sent the packet $P(i)$; otherwise, if $\text{FLAG}$ is FALSE, it denies. The attack node $x$ cannot forge the response to impersonate node $u$ because it does not have the pairwise key $K_{vu}$; thus, it will not be able to pass the inspection. However, the attack node $x$ may maliciously drop the $\text{CHALLENGE}$ message pretending the message was lost in transmission. In this case, after a timeout, node $v$ increases the probability $p_c$ to 1 and starts to resend the same $\text{CHALLENGE}$ message whenever receiving a packet from a claimed source $u$. Then node $v$ starts to count challenge failure times; once the challenge failure times exceeds a threshold or the failures remain after a timeout, node $v$ drops the neighborship of node $u$ and further forensic analysis may be taken to identify the attacker. Contrarily, if node $v$ receives an $\text{ACK}$ message from node $u$ before the challenge failure times reaches the threshold, then node $v$ resets $p_c$ and clears the challenge failure times. Note that node $v$ must not change the $\text{CHALLENGE}$ message in the following challenges because the renewed $\text{CHALLENGE}$ message may give the attack node $x$ a chance to regain the trustworthiness of $v$ by getting the credential from node $u$ to response the new challenge. Moreover, node $v$ should not actively resend the $\text{CHALLENGE}$ message to node $u$ in order to avoid the attack node $x$ luring node $v$ into consuming network resource.

In this way, the neighborship verification process is symmetric in the sense that each of the two nodes will be convinced that the other is its current neighbor. Hence, the number of verification processes is reduced by one half. Also, the choice of $p_c$ should make a trade-off between security and performance. A larger $p_c$ leads to stronger security, but it incurs larger overhead due to the exchange of challenges and responses. To control the probability $p_c$ that a node receives a challenge, every neighbor sets its probability to challenge the node as $p_c = p_r/d$, where
where \( d \) is the estimated network node density. Finally, this scheme requires a node to buffer the MACs of several packets it has recently transmitted to answer possible challenges.

3.5 Scheme II: a location-aware verification scheme

In this scheme, instead of challenging each other randomly, two nodes do not verify their neighborhood when they believe they are highly likely in neighborhood, thus further reducing message overhead. This scheme assumes that every node knows its own current location and velocity because of a GPS, and the transmission range \( r \) of a legitimate node is a fixed system parameter known to all the nodes in the network.

Let node \( u \)'s current coordinate and its velocity be \((X_u, Y_u)\) and \( \vec{V}_u \), respectively. When it bootstraps its key chain commitment to a neighbor \( v \), it also sends its location parameter \( LP_u = (X_u, Y_u, V_u) \) to \( v \) in an authenticated way.

\[
\begin{align*}
\text{Without loss of generality, let node } v \text{'s location parameter be } LP_v = (X_v, Y_v, V_v) \text{ and its most recently disclosed key in its key chain be } K_v(i). \text{ Node } v \text{ responds with the following message.}
\end{align*}
\]

\[
\begin{align*}
v \to u : LP_v, i, \{K_v(i)\}_{K_v}, MAC(K_v(i) || K_v(i) || LP_v)
\end{align*}
\] (3-6)

After this message exchange, both \( u \) and \( v \) know the location parameter and the previous one-time authentication key from each other. An attacker cannot replay the above message since it cannot fabricate the next cluster key chain. In addition, \( u \) and \( v \) record the message exchange time and estimate the time when they will move out of each other’s transmission range. For
instance, node v records the time when it receives the response message from u as $t^0_u$ and the estimated time that node u will move out of its transmission range $t^1_u$. It keeps the time period $[t^0_u, t^1_u]$, referred to as Radio Effective Duration (RED), during which node u is very possible to remain within its transmission range. The RED is maintained as part of the record corresponding to node u in the trust list of node v.

The problem remained is how to evaluate $t^1_u$. Consider Figure (3-3, which depicts a snapshot of the moment when node u and node v are exchanging their location parameters. Node u is moving in velocity $\vec{V}_u$ and node v is moving in $\vec{V}_v$. To simplify this problem, we focus on node u and translate the coordinate such that node u stays still and node v is moving in a relative velocity $V_{vu}$ under this new coordinate. Our goal is to find out $d_0$, which is the distance between node v and the transmission boundary of node u along the extension line of $\vec{V}_{vu}$.

![Figure 3-3: A snapshot of nodes u and v moving at velocities $\vec{V}_u$ and $\vec{V}_v$, respectively. Dashed circles indicate the transmission range of the two nodes. The analysis is used to predict when the two nodes will move out of the transmission range of each other.](image)
Consider the triangle $\Delta A_{uv}$ in Figure 3-3. Since we already know the transmission range $r$ and the distance $d_{uv}$, we can acquire $d_0$ by applying Equation (3-7).

$$r^2 = d_0^2 + d_{uv}^2 + 2d_0d_{uv}\cos\theta$$  \hspace{1cm} (3-7)

where $\theta$ is the degree of the angle on vertex $v$ in $\Delta A_{uv}$. The value of $\cos \theta$ can be obtained from the following equation.

$$\cos \theta = \frac{|\vec{v}_{uv}|}{|\vec{vu}|}$$  \hspace{1cm} (3-8)

where $\vec{vu}$ is a relative position vector pointing from $v$ to $u$ and $|\vec{vu}| = d_{uv}$. By rearranging Equation (3-6), we deduce a quadratic polynomial shown in Equation (3-9). By solving this equation, we can get two roots, which represent $d_0$ and the distance from $v$ to the transmission boundary of $u$ along the opposite direction of $\vec{V}_{uv}$ respectively.

$$d_0^2 + 2d_{uv}\cos \theta d_0 + (d_{uv}^2 - r^2) = 0$$  \hspace{1cm} (3-9)

After getting $d_0$, the time $t_u^1$ can be calculated by dividing the $d_0$ by the relative speed $|\vec{V}_{uv}|$ as the following equation:

$$t_u^1 = t_u^0 + \frac{d_0}{|\vec{V}_{uv}|}$$  \hspace{1cm} (3-10)

Once calculating its RED for node $u$, node $v$ will use it to decide whether or not to challenge node $u$ when it receives packets from node $u$ later.

To make the scheme as general as possible, we have estimated REDs based on the current location parameters of mobile nodes. In practice, however, the estimated REDs may become invalid due to the rapid change of node velocities. Note that this inaccuracy does not introduce
new security vulnerability if two nodes have moved out of each other’s transmission range before their REDs have expired. However, if they are still in each other’s transmission range after their REDs have expired, they should not discard the data packets from each other immediately; otherwise, a large number of legitimate packets will be falsely dropped. To address this issue, we adopt a buffering strategy, in which a node first temporarily buffers a data packet from a node whose RED has expired and then challenges that node for re-authentication, thus minimizing the impact of RED estimation errors on data delivery ratio.

Now we show the basic steps involved in the authentication process. Suppose node $v$ receives a data packet authenticated with node $u$’s one-time authentication key $K_u(i)$, it will perform one of the following steps:

Step 1. If the current time is between $[t_u^0, t_u^1]$ and to its knowledge that $K_i$ has not been released yet and $K_i$ is valid, it accepts the packet and passes the packet to the routing protocol.

Step 2. If the current time is larger than $t_u^1$, it temporarily buffers the packet. In addition, it marks the sender $u$ expired, moves $u$ from its trust list to a pending list, and then verifies its neighborhood with node $u$. The messages exchanged between them are similar to that in Equations (3-5) and (3-6) except that they both exchange the most recently released one-time authentication keys. After successfully exchanging their messages, node $u$ and node $v$ update their REDs and renew their trust relationship. Finally it goes to Step 1.

Step 3. If it challenges node $u$ but no reply has been received within a threshold time, it either drops the packet and removes node $u$ from its pending list or challenges node $u$ again with probability $p_2$ (here $p_2$ may decrease with the number of previously failed challenges to mitigate active attacks). If a challenge succeeds, it updates the corresponding RED and goes to Step 1.
Figure 4 depicts the state transition diagram used by node v when it receives different types of packets from node u. In the diagram there is no transition from ‘non-trust’ to other states, which indicates that any packets from a revoked node will be dropped immediately. However, if the enforced security policy allows a revoked node to come back for whatever reasons, the network controller could broadcast a re-authorization message. In this case, a transition from ‘non-trust’ to ‘unknown’ may be added.

This challenge and acknowledgement activity can be taken as a kind of synchronization. However, the frequency of synchronization in LIP depends on the relative mobility between the two nodes. As a result, if the relative positions of the two nodes are small, either moving in the same direction or in slow speeds, there would be a long period that the two nodes have no need to synchronize. Thus, we take the advantage of saving the control overhead. In some cases the relative speed of two nodes may be too small, causing $t_u$ in Equation (3-10) to become very large. For security reason, we set an upper bound time period RED₀ on RED. Two nodes will verify their neighborship within RED₀ even if their estimated RED is larger than RED₀.
Finally, we mention that there exists a DOS attack against the above scheme because of the buffering strategy. For instance, an attacker may impersonate a legitimate node \( u \), which has moved out of neighborhood and whose RED has also expired, by injecting a large number of data packets to node \( v \). Given a fixed buffer space, this attack could cause node \( v \) to drop data packets from other legitimate neighboring nodes whose REDs have expired. To address this attack, node \( v \) could allocate a fixed buffer size for each pending node and enforce a drop-head or drop-tail packet dropping strategy. If the number of buffers is also limited, node \( v \) could first recycle the buffers for the nodes who were challenged earlier but still have not responded yet.

The above attack seems to also result in service denial to node \( u \). Because node \( v \) challenges the impersonated node \( u \) with a decreasing probability \( p_2 \) after the first challenge message, when node \( u \) moves into the neighborhood again later, many of its legitimate data packets may be dropped. However, this situation does not really happen. When node \( u \) moves into the neighborhood, it will first resynchronize with node \( v \) by sending a challenge message, which node \( v \) will process immediately. If the challenge message is valid, node \( v \) will move \( u \) to the ‘trust’ state, thus no data packets will be dropped by mistake.

### 3.6 Further discussions

#### 3.6.1 Interaction with routing protocols

LIP is independent of the (secure) routing protocols. It requires that a node use its one-time authentication keys to authenticate all its packets to its direct neighbors despite the type of transmission (e.g. unicast, multicast or broadcast) of each packet. In practice, it could take advantage of the deployed routing protocol to provide stronger security. For example, if LIP can infer from the header of the routing protocol that a (data) packet is to be unicast (e.g. in a unicast-
based application), it can use its pairwise key shared with the next hop to authenticate the packet because using pairwise key can prevent impersonation attacks. On the other hand, since the security services provided by LIP are complementary to those provided by secure routing protocols, they can be employed at the same time to provide stronger security.

3.6.2 Key chain generation and renew

Subject to the network traffic patterns and the network lifetime, a node may need to transmit (forward or originate) a very large number of traffic packets. This will consume a large number of one-time authentication keys because every authentication key is only used once. The issue of providing a sufficient number of one-time keys has been addressed recently based on techniques such as Merkel-hash tree and multilevel key chains [15][26]. We note that a key chain may need to be discarded without being used up upon a node revocation. When a node $u$ knows that another node $v$ is being revoked (e.g. announced by a trusted authority), if node $v$ has been its neighbor and knows one or some of its one-time authentication keys, node $u$ should discard any future keys in its key chain and bootstrap a new key chain to its current neighbors other than $v$ (e.g. GKMPAN [44]). This completely prevents node $v$ from impersonating $u$.

3.6.3 Legacy issue

Scheme II requires every node to be equipped with a GPS; this requirement may not be easily met in the very near future for every application of ad hoc network. To support incremental deployment, we consider the case where only a fraction $\alpha$ of nodes is equipped with GPS devices (referred to as GPS nodes) while the rest do not have GPS devices (referred to as non-GPS
nodes). Clearly, Schemes I and II are special cases of this hybrid scheme when $\alpha = 0$ and $\alpha = 1$, respectively.

The coexistence of GPS nodes and non-GPS nodes poses several questions. Firstly, how can a node distinguish these two kinds of neighbors and then respond with the correct control message? The way to distinguish between GPS and non-GPS nodes is to incorporate one flag bit in each packet telling which type of node the sender belongs to. Secondly, which scheme should two nodes use to authenticate their packets? They run Scheme II (with the location-aware verification) only when they both are GPS nodes; in other cases they apply Scheme I (with the random neighborhood verification process) due to the lack of information to compute their distance and relative velocity. As such, $\alpha^2$ fraction of node pairs run Scheme II and $(1 - \alpha^2)$ fraction of pairs run Scheme I. In addition, the recorded information in a node’s trust list for two types of neighbors is also different.

3.6.4 Potential applications

The per-hop per-packet authentication capability provided in LIP is not only critical for building any applications that are open to attackers, but also necessary for many civilian applications of ad hoc networks that do pricing or require cooperation stimulation and incentive [11][27][40]. To promote cooperation, a node must forward packets for other nodes to receive credits to enable the forwarding of its own packets. For example, in Buttyán and Hubaux [11], a node is required to pass each packet to its security module. The security module maintains a counter, which is decreased when the node wants to send a packet as originator and increased when the node forwards a packet. The value of the counter must remain positive so that the node can send its own packets. Clearly, the secure module or the similar stimulation mechanisms can
make use of our per-packet authentication protocol to prevent selfish nodes from getting free rides. Hence, our mechanisms might be utilized as an enforcing technique for the system.

Another potential application is for providing traceback services in mobile ad hoc network. IP traceback has been extensively studied as a way to identify the attack source that launches distributed DoS attacks [37][35][36]. The basic idea is to mark a packet probabilistically [36] or record a digest of a packet [37][35] such that the information of the intermediate links or nodes is preserved and recoverable, thus tracing back to the source step-by-step. Here a fundamental security assumption in most of these schemes is that the intermediate routers are cooperative so that the reconstructed link or node information is trustworthy. While this assumption is generally believed to hold in the internet environment, in mobile ad hoc networks with malicious nodes the per-link information can no longer be assumed unless security mechanisms such as our schemes are employed to enforce per-link authenticity. Although traceback in mobile ad hoc networks is still an open problem, our mechanism at least provides a valid building block for the future research on this topic.

3.7 Security analysis

This section analyses the security of our schemes based on the security threat models in Hu et al. [21]. The security threat of these models, from Passive to ActiveCCX, increases with strength.

- **Passive**: an attacker, without cryptographic keys from the network controller, only passively eavesdrops on the traffic. As long as the underlying encryption algorithm and authentication algorithm are secure, an attacker cannot break the one-time authentication keys of the legitimate nodes.
• **Active I:** an active attacker attempts to inject malicious packets into the network although it has no cryptographic keys from the network controller. Since our protocol performs hop-by-hop authentication of every packet, without knowing a valid one-time authentication key, the attacker cannot inject its own packets into the network. Therefore, our schemes can prevent the ActiveI attack. We note that it is possible that the attacker replays another node’s packets, however, the attacker achieves little by doing this. This is because

- The attacker can only replay the packets transmitted by a node u to the other nodes that possess node u’s one-time authentication keys.
- If the nodes having node u’s one-time authentication key are node u’s current neighbors, they will drop the duplicated packets based on packet sequence numbers or one-time authentication key versions. For example, if an attacker replays node u’s ROUTE REQUEST packets to node u’s neighbors in DSR [23], the replay attack does not lead to multiple flooding of the same packet in the entire network because of the request id in the packet.
- If the nodes possessing some one-time authentication keys previously released by node u are not neighbors of u, they may accept the packet if the packet is a broadcast packet and they have not received it before. This however actually increases the reliability on broadcast messages.
- If time synchronization is provided (e.g. using GPS in Scheme II), a timestamp can be used to further prevent replay attacks.

• **ActiveX:** this threat model consists of multiple instances of the ActiveI model. An ActiveX attack is not more severe than a single ActiveI attack for LIP except when multiple attackers collude to launch a wormhole attack [21]. Scheme I cannot prevent the wormhole attack. In Scheme II, two neighboring nodes exchange their location parameters once they receive the
first packet from each other, whereby determining their RED, say $[t_0, t_1]$. If they are in neighborhood before $t_1$, they will be able to detect and prevent the wormhole attack. However, if they move out of each other’s transmission range at $t_m$ before $t_1$, they cannot completely prevent the attack during $[t_m, t_1]$. We note that this vulnerability can be addressed by letting a node include its current location in each packet, but this incurs larger bandwidth overhead.

- **ActiveC**: one active attacker node has all the cryptographic keys of a compromised node. Since an ActiveC attacker takes over the node, it can do whatever a node is allowed to do in the system on behalf of the compromised node. For most security systems, we have to resort to using some intrusion/misbehavior detection techniques [30][42] to defend against this type of attack. Another attack an ActiveC attacker can launch in our protocol is the impersonation attack due to the use of MAC-based broadcast authentication schemes. Both Schemes I and II are designed to mitigate this attack.

- **ActiveCX and ActiveCCX**: in the ActiveCX threat model, multiple active attacker nodes have all the cryptographic keys of one compromised node, whereas in the ActiveCCX model, multiple active attackers have all the keys of multiple compromised nodes. To reduce the risk of being detected because of using the same identity, ActiveCX attacker nodes are usually distributed in different locations of the network. ActiveCCX attacks can be even more sophisticated and difficult to detect. Our protocol alone does not have a solution for addressing this attack. We note a better solution is that every node is installed with an Intrusion Detection System (IDS). Moreover, multiple nodes could also perform cooperative detection, for example, by recording and exchanging their neighborship information. Zhang and Lee [42] and Marti et al. [30] have studied the intrusion and misbehavior detection issue
in mobile networks. We believe this is still an open area and a study that identifies the possible attack patterns is the first step towards addressing these attacks.

3.8 Performance analysis

The goal of our experiments is to measure the performance overhead introduced by LIP when the network is not under adversary attack. In particular, we desire to respond to the following questions: How much bandwidth overhead does LIP introduce? How many packets does LIP drop? What is the impact of node mobility model? Finally, what is the impact of location-awareness on LIP? The simulation results are based on Scheme II, the location-aware scheme, unless otherwise noted.

3.8.1 Metrics

We mainly consider the following performance metrics in this evaluation:

- **Control overhead**: we define control overhead as the transmission overhead (in bytes per second per node) introduced by the proposed scheme. This includes one MAC attached to each packet and all the challenge-response messages in the protocol.

- **Traffic delivery ratio**: we define the traffic delivery ratio as the fraction of traffic packets that a node accepts to the total number of packets it receives from its legitimate neighbors. The higher the traffic delivery ratio, the smaller impact on the upper layer protocols.

Note that here we do not consider computational overhead because the scheme mainly involves several symmetric key operations (MAC and hash computations), which are computationally efficient. The other computational overhead would be the infrequent estimation of REDs, which is very fast to compute.
3.8.2 Simulation methodology

The simulation utilizes GloMoSim 2.02 [17] and the following default configuration: There are 100 nodes distributed in a square environment space of 2000 m × 2000 m. Each node joins the network at a time uniformly distributed between the simulation time 0 and 5 sec and each simulation runs for 900 sec of simulated time. In physical layer, the radio propagation model is two-ray ground reflection model. In medium access control layer, we used IEEE 802.11 Distributed Coordination Function (DCF). To demonstrate that LIP is independent of the routing protocol, we inserted LIP beneath three different routing protocols: the unicast routing protocols DSR [23] and AODV [32], and the multicast routing protocol ODMRP [28]. When the routing protocol is either AODV or DSR, we picked up 13 source-destination pairs for unicast communication. The setting for ODMRP is that, of 100 nodes, 13 nodes form one multicast group and 12 nodes form another multicast group. The rest of 75 nodes do not belong to either of these two groups. The rest of the parameters are simply the default values employed in GloMoSim.

In application layer, the traffic pattern is Constant Bit Rate (CBR). The size of a CBR packet is 512 bytes. We varied the intervals between two CBR packets from 0.1 to 1.0 sec and the durations of connections from 10 to 850 sec which reflect from the burst traffic pattern to the uniform traffic pattern in the network.

Table 3-1 summarizes the default parameters used in our simulation unless otherwise mentioned. Most of the settings are commonly used in the research literature [19][20][43].
Table 3-1: Simulation parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical link bandwidth</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Radio frequency</td>
<td>$2 \times 10^9$ Hz</td>
</tr>
<tr>
<td>Radio transmission power</td>
<td>15 dBm</td>
</tr>
<tr>
<td>Radio transmission sensitivity</td>
<td>-91 dBm</td>
</tr>
<tr>
<td>Radio transmission threshold</td>
<td>-81 dBm</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Waypoint</td>
</tr>
<tr>
<td>RED$^0$</td>
<td>200 sec</td>
</tr>
<tr>
<td>Threshold time for waiting responses</td>
<td>2 sec</td>
</tr>
<tr>
<td>Random neighbor challenge probability $p_c$</td>
<td>0.1</td>
</tr>
<tr>
<td>Rechallenge probability $p_z$</td>
<td>0</td>
</tr>
<tr>
<td>Location information size</td>
<td>8 bytes</td>
</tr>
<tr>
<td>Velocity information size</td>
<td>8 bytes</td>
</tr>
<tr>
<td>Time for MAC verification</td>
<td>1 $\mu$s</td>
</tr>
<tr>
<td>HMAC key size (including a key id)</td>
<td>10 bytes</td>
</tr>
</tbody>
</table>
3.9 Evaluation results

3.9.1 Computational overhead and latency

In LIP, a node normally verifies a received traffic packet by computing one HMAC and one hash, the time for which is less than 1 ms even for handheld PDAs [110] with very constrained computational capability. On an AMD Opteron 1.7 GHz processor, a hash over 8 bytes can be computed in about 0.08 μs [111]. The other computational load would be the infrequent estimation of REDs, which is negligible.

The processing delay of a packet at one hop (node) includes the time spent in all the layers. The time spent in transport layer or application layer is application dependent. Let us just consider the time spent in the medium access control layer and network layer while ignoring the computational overhead. The main processing time spend on the random back-off and the jitter in both the media access protocol and the routing protocol. In AODV, the average random back-off time is 250 ms and the average broadcast jitter time is 5 ms; in the 802.11 medium access control protocol, the average random back-off time is 25 ms. Thus, the delay caused by LIP is several-order smaller in comparison to the delay introduced in each hop.

3.9.2 The impact of node mobility

1.) Control overhead: Figure 3-5 shows that control overhead increases with node mobility. The control overhead is normally between 10 and 20 bytes/sec/ node for all three routing protocols. This indicates that our scheme has low bandwidth overhead.

The impacts of node mobility on control overhead are three folds: Firstly, with higher node mobility, RED is normally smaller. In other words, nodes have to synchronize with each other more frequently, which in turn increases the number of challenge and ACK messages.
Secondly, with higher mobility, a node will encounter more nodes. Chances are many of them have never been contacted before or their neighborship has expired. When a node receives a packet from its neighbor that is not in its trust list, it will automatically send a challenge to the neighbor and the neighbor will also respond with an ACK message. This round of challenge and ACK alone costs 58 bytes of control overhead. Finally, the increased node mobility requires more control packets to maintain network connectivity. Since LIP authenticates every packet, the increased amount of control packets also implies the enlarged per-packet associative overhead. Hence, the routing protocol which incurs less control packets also suffers less overhead imposed by LIP. DSR has lower control packet overhead than AODV and ODMRP. The overhead of LIP with DSR is also the lowest.

2.) Traffic Delivery ratio: Figure 3-6 depicts that the traffic delivery ratio of LIP is close to 1.0 and reduces slightly as the node mobility increases. In the worst case in DSR, a node drops about five traffic packets out of 10,000 packets it receives. The packet loss is mainly due to normal
network packet loss when the network topology changes. Another source of packet loss is dropping packets from a neighbor which failed to respond to a challenge. Since our scheme only makes proximate estimation of RED, it is possible that the RED of a neighbor node expires while it is still within the transmission range. In this case, traffic packets will be temporarily stored, a challenge will be sent, and after receiving an ACK from the neighbor, the traffic packets are passed to the routing protocol. If the neighbor moves out of transmission range before it can respond to this challenge, all of the packets temporarily stored will be dropped. We can also notice that DSR has slightly lower traffic delivery ratio than the other two. This is mainly because of the feedback implosion problem of the DSR protocol. In DSR a node may keep silent most of the time. As a result, it is possible that many of its neighbors be unaware of its existence until it broadcasts a message. In this scenario, the network traffic increases out of proportion since many neighbors will challenge it at the same time and hence a chance of losing more challenge packets.

Figure 3-6: Impact of node mobility on traffic delivery ratio.
3.) **Buffer Size:** Figure 3-7 depicts the temporary storage cost of LIP in terms of the number of packets. In this figure, the maximum buffer length represents the largest storage space that a node had ever used throughout the simulation run. As the node mobility increases, we can make two observations from the figure: Firstly, the average and maximum buffer length grows with the node mobility. The reason is that as the node mobility increases, a node encounters more neighbors. As noted earlier, a packet from a neighbor with an expired RED is temporarily stored, an ACK from it is received, and verified. Secondly, LIP under DSR has a lower storage overhead than AODV or ODMRP. This is because, under DSR, LIP has smaller trust lists. In the worst case, as shown in Figure 3-7, about 135 packets are buffered. This accounts for 67.5 kB. For many wireless devices such as PDAs, this storage overhead is quite reasonable.

![Figure 3-7: Impact of node mobility on the length of buffer.](image-url)
3.9.3 Impact of node density

To examine the impact of node density, we varied the number of nodes in the 2000m × 2000m field from 60 to 160. We also increased the number of source-destination pairs (approximately) linearly.

1.) Control overhead: Figure 3-8 shows the impact of node density on control overhead. As expected, we observed that the control overhead increases with node density under every routing protocol. This is because the control overhead of LIP depends on the communication load. The MAC attached to each packet for authentication accounts for the majority of the control overhead.

![Figure 3-8: Impact of node density on control overhead (node mobility 10 m/s).](image)

2.) Traffic Delivery ratio: Figure 3-9 indicates that the traffic delivery ratios are above 99.85% for all three routing protocols when the number of nodes is 160 or less. As shown in this figure, the delivery ratio for DSR is slightly reduced (notice that the scale is very small) because the source routing protocol (e.g., DSR) lacks of local route recovery mechanism in contrary to the vector based routing protocols (e.g., AODV and ODMRP) when a link within a route breaks up.
3.9.4 Impact of location awareness

As we discussed earlier, to support incremental deployment, we need to consider the case where only a fraction $\alpha$ of nodes are equipped with GPS facility while the rest are non-GPS nodes. In a separate simulation run, we examined the impact of $\alpha$ on the performance of LIP. Figure 3-10 illustrates that with challenge probability $p_c = 0.1$, for AODV and ODMRP, control overhead decreases with $\alpha$. DSR has a relative higher control overhead than the other two because each data packet attached entire route list in DSR rather than next hop in the other two. This is due to the fact that instead of randomly challenging a neighbor, a location-aware node knows more precisely when to challenge a neighbor based on its RED. The control overhead in LIP under ODMRP is relatively large, 50 bytes/sec node, because of the large number of control packets involved in ODMRP. To reduce the control overhead, one may reduce $p_c$ as a tradeoff.
between security and performance. On the other hand, our simulation results show that the traffic delivery ratios are above 99.99% under all the $\alpha$ values.

3.9.5 Impact of mobility models

In LIP, node mobility model affects REDs. For example, nodes moving in a group will have on average larger REDs (although bounded by RED$^0$) because their relative speeds are smaller. To examine the impact of mobility models, we introduced two mobility models: Reference Point Group Mobility (RPGM) [18] and Manhattan mobility model [13]. We compared these two mobility models with the default random waypoint mobility model at a maximum speed of 10 m/s. We utilize BonnMotion [12] to generate mobility scenarios. Table 3-2 and Table 3-3 list the parameters used to generate scenario files.
Table 3-2: The mobility model *RPGM* parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Average number of nodes per group</td>
<td>20</td>
</tr>
<tr>
<td>Group number</td>
<td>5</td>
</tr>
<tr>
<td>Group change probability</td>
<td>0</td>
</tr>
<tr>
<td>Maximum distance to group centre</td>
<td>50 m</td>
</tr>
<tr>
<td>Minimum speed</td>
<td>1 m/s</td>
</tr>
<tr>
<td>Maximum speed</td>
<td>10 m/s</td>
</tr>
<tr>
<td>Maximum pause</td>
<td>30 sec</td>
</tr>
</tbody>
</table>

Table 3-3: The mobility model *Manhattan* parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Minimum speed</td>
<td>1 m/s</td>
</tr>
<tr>
<td>Mean speed</td>
<td>10 m/s</td>
</tr>
<tr>
<td>Maximum pause</td>
<td>30 sec</td>
</tr>
<tr>
<td>Pause probability</td>
<td>0.001</td>
</tr>
<tr>
<td>Turn probability</td>
<td>0.2</td>
</tr>
<tr>
<td>Update distance</td>
<td>10 m</td>
</tr>
<tr>
<td>Number of blocks along $x$-dimension</td>
<td>10</td>
</tr>
<tr>
<td>Number of blocks along $y$-dimension</td>
<td>10</td>
</tr>
</tbody>
</table>
In RPGM, nodes are divided into logical groups. For each group, there is a reference point. All of the nodes belonging to a group move according to their reference point. Since the nodes of one group generally move together around, one should expect more stable neighborship among nodes in a group. This results in larger REDs. Figure 3-12 confirms this expectation. In Manhattan mobility model, although some geographic restrictions are imposed on node mobility,
we find that its REDs are the smallest because the vehicles moving on the road have strict trajectory; they rarely stick to each other or form a group. The control overhead in different mobility models, as shown in Figure 3-11 conform to the observations on the different REDs shown in Figure 3-12.

Overall, the above performance analysis shows that LIP is a lightweight protocol and it provides traffic delivery ratio close to 100% with different node speeds, network densities, and mobility models.

3.10 Conclusion

We have presented LIP, a lightweight hop-by-hop authentication protocol for preventing unauthorized nodes from injecting spurious packets into ad hoc networks. The protocol is transparent to and independent of the network routing protocols. The protocol relies solely on symmetric key operations and is based on a localized broadcast authentication technique whereby each packet is efficiently authenticated at each hop by the immediate neighbor(s) of the node transmitting the packet. In addition to being able to prevent resource consumption attacks by unauthorized nodes, the protocol can mitigate impersonation attacks by the compromised nodes. We proposed three different schemes for achieving this goal. Our performance analysis showed that the protocol has low communication overhead and high traffic delivery ratio.
Chapter 4

Online Traceback in Mobile Ad Hoc Networks

4.1 Introduction

TCP/IP was not designed to authenticate packet source addresses. As a result, a sender can assign an arbitrary source address to its data packet to deceive the receiver; the malicious exploitation of this protocol’s weakness is referred to as the IP spoofing attack. IP spoofing attack is generally coupled with other detrimental attacks in order to disguise the real attack sources. For example, the denial-of-service (DoS) or the distributed denial-of-service (DDoS) attacks flood the network with address-spoofed packets.

The increased frequency of DoS/DDoS attacks has motivated the development of many IP traceback protocols for the Internet [4, 5, 14, 10, 18, 20, 13, 24]. Among these, two types of IP traceback schemes dominate the literature: probabilistic packet marking (PPM) [8, 17, 20] and hash-based logging [18, 23, 10]. The basic idea of a PPM scheme is that an intermediate node probabilistically mark its node/link information into a packet during forwarding, consequently a victim may gather packet mark from each intermediate node. Based on the gathered information, the victim may reconstruct the entire attack path after it has received a sufficient number of marked packets. A hash-based logging scheme works by requiring an intermediate node record the hash of a forwarded packet (or use Bloom filter [3] to further reduce storage overhead), such that later on an investigator may poll the intermediate nodes with the hash to reconstruct the attack path.

IP traceback has been extensively studied in the literature, however, within the scope of mobile ad hoc networks (MANETs), the research was recently attempted to define traceback
protocols [2, 9]. In MANETs, the network nodes forward packets in a self-configuration and self-maintenance purposes without any infrastructure support. While both the scale of a MANET and its data traffic rate are much smaller than its high-speed Internet counterpart, nevertheless, online (or real-time) traceback in MANETs imposes some unique challenges that are uncommon in the Internet:

- Node mobility is one key factor. In a MANET, the network topology is constantly changing because the wireless link dynamically built up when two mobile nodes move into each other’s radio transmission range. The dynamic network topology fundamentally changes the paradigm for attack source traceback. The existing IP traceback schemes may not work in MANETs, because almost all of the IP traceback schemes assume a static network topology.

- Trust is another key factor. Unlike the Internet where routers are often trusted, the mobile nodes in MANETs normally should not be trusted. Consequently, a traceback protocol for MANETS itself suffers from malicious attacks.

One may propose a new traceback protocol by modifying an existing protocol based on the aforementioned limitations and characteristics of the MANETs. However, such a solution should answer the following question: how will such a scheme performs in MANETs? To answer this question, we quantitatively analyze the impact of node mobility on the performance of two representative types of traceback schemes (i.e., PPM and logging schemes). Specifically, we formulate the impact of the network parameters (e.g., the length of an attack path, the victim response time, and the mobility) on the traceability of these schemes in MANETs. Our analytical results show that (i) the traceability of both schemes decrease as node mobility increases, and (ii) a PPM scheme is vulnerable to low-rate attacks, while a logging scheme performs poorly when a victim has a relatively high intrusion response time.

We then propose a new online traceback scheme for MANETs. Our scheme is capable of detecting a hotspot where a malicious attack resides. Based on the principle of divide and
conquer, our scheme works by dividing a forwarding path into multiple smaller interweaving fragments. During a traceback process, those reachable fragments are reconstructed and fragmentation information is gathered. The proposed solution consists of three novel building blocks: verifiable and distance-based packet marking, multicast-based traceback query forwarding, and hotspot-based attacker detection. The first building block can prevent various insider attacks that are often effective against conventional PPM schemes; the second building block can bypass some broken links due to mobility; the third building block reports the minimum hotspot by analyzing the traceback results. Through simulations in various settings, we show that each traceback that employs our scheme can attribute to a much smaller hotspot where the attack source resides than a conventional logging scheme may provide.

4.2 Preliminaries

4.2.1 Network model and security assumptions

We consider a traditional mobile ad hoc network (MANET) where nodes form a network on-the-fly and forward packets for one another. At the initial stage of our research, we assume that all of the network nodes in the MANET employ the same traceback protocol. Furthermore, they can establish trust through either a PKI, a Trusted Third Party (TTP) or predistributed shared keys [11, 25]; thus, any two nodes can establish a pairwise shared key for packet authentication. During data forwarding, every packet is authenticated in a hop-by-hop fashion [27]; that is, the link between two neighboring nodes is assumed to be authenticated and a malicious node cannot impersonate any good node.
For a node $u_i$ located on an attack path between the source $S$ and the victim $V$, a node $u_j$ is called its upstream node if $u_j$ is closer to $S$ than $u_i$ is. Similarly, $u_j$ is a downstream node of $u_i$ if $u_j$ is closer to $V$.

### 4.2.2 Attack model

We assume the adversary may compromise one or multiple nodes and take full control of the compromised node(s). The adversary may launch an attack against a specific victim node via a single compromised node, coordinate multiple compromised nodes to simultaneously flood the victim, or make the attack source and the compromised nodes conspire to conceal the attack source. Since a compromised node possesses a valid security credential, the injected packets will not be detected by its forwarding neighbor or the other intermediate nodes. A compromised node may also alter or drop any received packets.

We consider two types of false packets: First, an injected packet may carry a malicious data payload that exploits a software vulnerability in the victim’s program to cause, for example, buffer overflow based remote code execution. If not updated with an appropriate security patch or virus signature, the victim may be damaged, taken over, or its sensitive information may be stolen. The size of an exploit code is normally not large, say from hundreds of bytes to tens of kilobytes [121], so for this attack, in order to effectively attack a victim, one to tens of packets will be sufficient. In the second case, the false packets are used for DoS attacks, consuming the bandwidth or computation power of a victim node. We will expect at least hundreds of packets for this attack to be effective.

In either of these cases, since the links are authenticated in a hop-by-hop manner, the attack source cannot impersonate any normal (benign) node to its downstream node. To hide itself, it will not put its address into the packet source field; instead, it will act as if it was a data
forwarder for the packets while spoofing valid source ids. Finally, the attack source may change its location over time to hide itself.

4.2.3 Design goals

In the Internet, an attacker source is a host, not a router, and routers are trusted, thus as long as the edge router of the attack source is identified, the source will be convicted. In other words, IP traceback is designed to identify the precise identity of an attack source assuming the intermediate routers are trustworthy. The same goal, however, is hard to achieve in a MANET because all nodes are equally trusted. When an intermediate node is compromised, it may collude with the attack source to deceive the traceback initiator and thus neutralize a traceback process. Also, whenever a dispute occurs between two nodes or among multiple nodes, a traceback protocol alone often cannot arbitrate the dispute.

Without loss of generality, consider an attack path $A_M$ of packet $M$ in Figure 4-1, where the source node $S$ and the intermediate node $b_3$ are compromised and are both at the disposal of the adversary. $b_3$ may alter packet markings (when a PPM scheme is in place) or drop traceback queries. If a traceback protocol can only trace to $u_4/b_3$ (e.g., $b_3$ drops the traceback query or traceback reply), then $S$ becomes invisible to the downstream nodes and the victim. If $b_3$ does not
interfere the traceback process, the protocol may reach $u_i/S$. In either case, $b_3$ or $S$ will probably deny its attacking behavior and consequently a dispute may arise between $b_3$ and $u_4$ or between $S$ and $u_1$. The traceback protocol, unlike in the scenario of the Internet, unfortunately, is unable to precisely identify the attack node.

As such, we set forth the following realistic design goals: First, we try to locate the hotspot where the compromised node resides. Hotspot-based traceback was previously introduced in [58]. Once a hotspot is identified, we will rely on other online or offline analysis/detection measures (e.g., neighbor watching [30]) or human intelligence to pinpoint the attacker. We assume hotspot analysis will eventually reach the real attacker given the authenticated links. Second, we aim to identify at least one of the malicious nodes—$S$ or $b_3$ in this example. From the attacker’s perspective, the exposure of any one of its controlled nodes may have the same impact on the capability of its future attacks.

While these are not remote design goals, it remains an issue on how to implement them most effectively. Hotspot analysis is often expensive to conduct and its cost increases with the size of the hotspot under investigation. In a MANET, the size of a hotspot, which is the number of hops from the attack source to where the traceback query reached, could be magnified by node mobility. For example, in Figure 4-1, due to mobility, node $u_4$ may become unreachable during a traceback. In this case, the victim is unable to determine whether $u_4$ moved away or it intentionally rejected the traceback query. Indeed, not only all the upstream nodes of $u_4$ including the malicious nodes $b_3$ and $S$ are invisible, but also $u_5$ falls in the hotspot, resulting in a hotspot of size 6. Clearly, hotspot based traceback protocol should consider minimizing the size of a hotspot. In addition, it should also minimize the number of packets required for a successful traceback, so that one could detect low rate attacks and catch the attacker as early as possible.

We assume the victim is the initiator of a traceback, although a trusted investigator, if existing, may be assumed to conduct a traceback on the behalf of the victim. The prerequisite of
applying the online traceback scheme is that the victim can detect the malicious intrusion in the first place. Therefore, it is reasonable to assume the victim has installed an appropriate intrusion detection system (IDS) [42]; otherwise, the argument would not be able to continue. Furthermore, the response time of the IDS should be small, say from seconds to tens of seconds; otherwise, any online traceback scheme would not be useful because the lifetime of a MANET could be very short, nodes may join and leave the network frequently, and node mobility may cause the network disconnection. Practically, the existing intrusion detection scheme based on worm signature [122] has feasible response time. Finally, we assume both traceback queries and reports are authenticated.

4.3 Traceability Analysis of Existing Schemes for MANETs

To design a traceback scheme for MANETs, we start from examining the feasibility of directly applying the existing IP traceback schemes to a MANET environment. Intuitively, due to the mobility as noted earlier, an IP traceback scheme is not well suited for MANETs. However, no concrete analysis to quantify such intuition has been reported in the literature. A quantified analysis will clearly show how the current IP traceback protocols are susceptible to the dynamic topology in MANETs and it will serves as a metric for evaluating any new proposed traceback scheme for MANETs. As such, we will first make a traceability analysis of existing IP traceback schemes before presenting our approach in the next section.

The common IP traceback schemes as advanced in the literature can be categorized into marking-based schemes and logging-based schemes. Therefore, our discussion below will be focused on these two approaches. We define traceability $\mathcal{I}$ as the success rate of traceback in MANETs; we call it a success when the immediate downstream neighbor of the attack source is identified.
4.3.1 Marking-based schemes

In a marking-based scheme, e.g., probabilistic packet marking (PPM) [37][55], routers probabilistically mark the packets being forwarded with partial path information, which later on allows a receiver to reconstruct the forwarding path given a modest number of the marked packets. In the edge sampling algorithm [55], an IP traceback mark consists of a distance field and a start-end pair. When a router decides, with a preset probability $p$, to inscribe a packet, it overwrites the start field of the packet with its own IP address and sets 0 into the distance field; if the distance field is already 0, indicating the previous router has marked the packet, the router writes its own IP address into the end field to create an edge between the previous hop and the current router in the packet. Finally, if the router is not going to mark the packet (with probability $1 - p$) and the distance field of the packet is not 0 (if the distance is 0, the router is the end of the edge and will inscribe a mark to the packet), it simply increments the distance field. This distance field, hence, represents the number of hops between the victim and the closest router that marked the packet.

For an attack path of length $d$, the victim can trace to the attack source only if it receives at least one packet marking from the immediate neighbor $u_1$ of the attack source before the path breaks up. Hence, the traceability $\mathbb{I}_{ppm}$ of PPM scheme for MANETs is determined by two factors: the packet rate and the path duration $PD$ given a fixed marking probability $p$.

Assuming that each packet carries only one marking at a time, and assuming that the routers are trustworthy, the expected number of packets $E(X)$ that the victim has to receive before receiving the marking from $u_1$ is:

$$E(X) = \frac{1}{p(1 - p)^{d+1}}$$

(4-1)
If the attacker sends the packets at a constant packet rate $\gamma$, then the expected time $T_{\text{mark}}$ by which the victim receives the marking from $u_1$ would be

$$T_{\text{mark}} = \frac{E(X)}{\gamma} \quad (4-2)$$

Sadagopan et al. [63] proposed a theoretical model to approximate the path duration based on the analysis of statistical of extensive simulation results. According to their analysis, path duration can be approximated by an exponential distribution when the network nodes move in moderate and high velocity. The exponential random variable has the following cumulative distribution function (CDF)

$$F_{PD}(t, d) = \begin{cases} 1 - \exp\left(-\frac{\lambda_0 d v}{R} \cdot t\right), & t \geq 0 \\ 0, & t < 0 \end{cases} \quad (4-3)$$

where $R$ denotes the radio transmission range, $d$ denotes path length, $v$ denotes the max velocity of a mobile node, and $\lambda_0$ is the proportionality constant.

We define the traceability as the probability that the path duration $PD$ is greater than $T_{\text{mark}}$. Thus, we may derive the traceability for PPM as

$$\Xi_{ppm} = 1 - F_{PD}(T_{\text{mark}}, d) \quad (4-4)$$

From Equation (4-3) and (4-4), we have

$$\Xi_{ppm}(d, \gamma) = \exp\left(\frac{-\lambda_0 \gamma}{R \cdot p(1 - p)^{\gamma - 1}} \cdot \frac{d}{\gamma}\right) \quad (4-5)$$

To pictorially demonstrate the limitation of applying PPM to MANET traceback, let us consider an example in which $R = 250m$, $v = 10$ m/s, $p = 1/20$, $\gamma = 1, 5, 10$ pkt/sec. Figure 4-2 shows the traceability of PPM scheme for various path lengths. From Figure 4-2 one can conclude that an
attacker for six hops away from the victim can launch an attack without being detected by controlling the packet rate at 1 pkt/s.

From Figure 4-2, one can also conclude that the traceability $\Im_{ppm}$ decreases with $d$ and $\gamma$ affect $\Im_{ppm}$ inversely. (Note that $v$, $R$, and $p$ are system parameters and fixed for a network configuration), i.e. $\Im_{ppm} = f(d, \gamma)$. Unfortunately, both the path length $d$ and the packet rate can be controlled by the adversary. The victim can basically do nothing to improve the traceability in the PPM scheme. Therefore, application of PPM traceback in MANETs is an unfair game

$$\min_{d, \gamma} \Im_{ppm}(d, \gamma)$$  \hspace{1cm} (4-6)

which overwhelmingly favors the attacker.
4.3.2 Logging-based schemes

In a logging-based scheme, every intermediate node records the message digest of a forwarded packet; thus, every packet leaves a trail on the path from its source to its destination. An attractive characteristic of logging-based schemes is that they can trace the attack source with as less as a single packet, as long as the entire path can be reconstructed. To reduce the storage overhead for keeping the message digests, a typical space-efficient data structure called Bloom filter [116] can be used. A practical architecture, Source Path Isolation Engine (SPIE) (also known as the hash-based traceback) was proposed in [65][66], in which the SPIE adapter is deployed on the intermediate routers to audit the traffic and digest the invariant portion of each packet for later queries. When the victim identifies an attack with its intrusion detection system (IDS), it launches a traceback query to a traceback agent. The traceback agent is authorized to poll each of the intermediate routers. Each polled router identifies the false packet trail by looking it up in its own Bloom filter and reports the result to the agent. The traceback agent rebuilds the attack graph with the help of the information about the network topology.

To conduct an on-line traceback for MANETs, we may employ the similar technique. The victim quickly responds to an attack by initiating a traceback process. The traceback query is expected to rapidly propagate along the reverse attack path, hoping that the traceback query can reach the neighbor of the attack source before the attack path breaks up.

Let us denote $\mathcal{A}_x$ as the attack path associated with the attack packet $x$,

$$\mathcal{A}_x = (S, u_1, u_2, \ldots, u_d, V)$$ (4-7)

where $S$ is the attack source, $u_i$, $i = 1, 2, \ldots, d$, are the intermediate nodes, and $V$ is the victim node. Without loss of generality, consider a spoofed packet $x$ is sent from $S$, through $u_1, u_2, \ldots$, to $V$. After processing the packet with intrusion detection algorithm, $V$ concludes that $x$ is a
malicious or a false packet, so it sends out a traceback request TBREQ toward the reverse path of \( A_x \).

Analogous to the marking-based scheme, the path duration \( PD \) also controls the traceability. Therefore, we consider the traceback behavior from two points of views: path duration and traceback latency. A traceback is successful when the traceback latency is less than path duration \( PD \). The traceback latency consists of the end-to-end propagation delays plus the victim IDS response time \( t_{resp} \). Figure 4-3 illustrates the unicast traceback model:

![Traceback Model Diagram](image)

Figure 4-3: The traceback latency. \( t_{sv} \) is the end-to-end propagation delay from \( u_1 \) to \( V \). \( t_{vs} \) is the end-to-end delay from \( V \) to \( u_1 \). \( t_{resp} \) is the victim node response time.

According to the traceback model, we define the unicast traceability \( \mathcal{I}_{\text{single}} \) with regard to the false packet \( x \) as

\[
\mathcal{I}_{\text{single}} = \Pr\{PD > t_{sv} + t_{vs} + t_{resp}\} \quad (4-8)
\]

Note that a traceback process will only be initiated after the victim node receives the false packet \( x \) successfully. Hence we can rewrite the traceability formula as

\[
\mathcal{I}_{\text{single}} = \Pr\{PD > t_{sv} + t_{vs} + t_{resp} \mid PD > t_{sv}\} \quad (4-9)
\]

As described in the previous section, \( PD \) can be modeled as an exponential random variable, and since the exponential random variable is memory less, Equation (4-9) can be reduced to
\[ \mathcal{I}_{\text{single}} = \Pr \{ PD > t_{VS} + t_{\text{resp}} \} \]  

(4-10)

Assuming the end-to-end delay is uniformly distributed and is proportional to the path length \( d \) (i.e., \( t_{SV} \propto d \) and \( t_{VS} \propto d \)), the unicast traceability \( \mathcal{I}_{\text{single}} \) will then be a function of \( d \) and \( t_{\text{resp}} \). Based on Equation (4-3), we have

\[
\mathcal{I}_{\text{single}} = 1 - F_{PD}(t_{VS} + t_{\text{resp}}) \\
= \exp \left\{ -\lambda_0 d \frac{v}{R} (k \cdot d + t_{\text{resp}}) \right\}
\]

(4-11)

where \( k \) is the average delay in an intermediate node. Clearly, \( \mathcal{I}_{\text{single}} \) decreases with \( v \) as the node mobility (i.e., \( v \)) increases. Similar to our discussion in Section 4.3.1, \( v \), \( R \), and \( k \) are determined by the physical characteristics of the platform. Hence, in the following discussion, we study the effect of \( t_{\text{resp}} \) and \( d \) on \( \mathcal{I}_{\text{single}} \).

Consider a MANET with the settings: \( v = 10 \text{ m/s}, R = 250 \text{ m}, k = 80 \text{ ms/hop}, \) and \( \lambda_0 = 0.7 \) (We used the method of Maximum-Likelihood [117] to estimate the constant \( \lambda_0 \) in Equation based on the samples of our simulation results). Figure 4-4 depicts traceability \( \mathcal{I}_{\text{single}} \) as the function of path length \( d \) and response time \( t_{\text{resp}} \). It is easy to see that traceability decreases quickly with \( d \) and \( t_{\text{resp}} \). To validate this model, we also ran simulations with similar settings, where the network nodes move in random waypoint mobility model. A comparison of the analytical results with the simulation results is also shown in Figure 4-4, which justifies the validity of our analytical model.
Compared with a marking-based scheme, the logging-based online traceback scheme is a more fair game. The victim node controls the response time $t_{resp}$ and the attacker only controls the length of the attack path $d$. Therefore, this online traceback is a min-max game.

$$\min_d \max_{t_{resp}} \mathcal{I}_{single}(d, t_{resp})$$

(4-12)

According to Equation (4-11), we notice that

$$\mathcal{I}_{single} \downarrow 0 \quad \text{as} \quad d \uparrow \infty$$

$\mathcal{I}_{single}$ unfairly favors the adversary when the path length $d$ is unbounded. However, we observe that the adversary also suffers from the long distance – the longer the attack path is, the more likely the attack packet will be lost due to the changes in topology. Therefore, we explore the relationship between the traffic delivery ratio $R$ and the maximum path length $d_{max}$ as follows:

Figure 4-4: Traceability as the function of path length. The dashed lines present the theoretical traceability and the solid lines present the simulation results. $T_{resp}$ denotes the response time of the victim node.
\[ R = \Pr\{PD > t_{sy}\} \]
\[ = 1 - F_{pd}(k \cdot d) \]
\[ = \exp\{-C_A \cdot d^2\} \] (4-13)

where constant \( C_A \) equals to \( \frac{\lambda_0 v_k}{R} \). Given a required minimum traffic delivery ratio \( R_0 \), we may derive the maximum path length \( d_{\text{max}} \) by adding the constraint

\[ R_0 \leq \exp\{-C_A \cdot d^2\} \]

where \( 0 \leq R_0 \leq 1 \), and \(-\ln R_0 \leq C_A \cdot d^2\). We may derive the range of path length

\[ 1 \leq d \leq \sqrt{-\ln R_0 \over C_A} \]. According to this inequality, we define the maximum path length \( d_{\text{max}} \) as

\[ d_{\text{max}} = \sqrt{-\ln R_0 \over C_A} \] (4-14)

Figure 4-5 plots \( d_{\text{max}} \) versus \( R_0 \) with the same setting used in Figure 4-4. For instance, to achieve 0.95 packet delivery rate, the adversary can only launch an attack as far as 6 hops away from the victim.
4.3.3 Further Discussion

In Section 4.3.1 and 4.3.2, we have quantitatively analyzed the traceability of the marking-based scheme and the logging-based scheme, when they are directly applied to MANETs. In general, applying marking-based schemes to MANET traceback encounters several new challenges, when considering both mobility and security: First, a victim node has to gather many packet marks to reconstruct the attack path, and hence it is incapable of detecting low-rate DoS attacks launched by a remote node. Second, the inscribed packet marks may not be genuine if intermediate nodes do not cooperatively abide by the marking protocol by altering or removing the marks. In other words, the authenticity of the edges in a reconstructed path cannot be verified. For logging schemes, its limitation lies in its susceptibility to node mobility. When a path is broken in a link, i.e., of the two nodes forming the link the one further away from the victim does not respond to a traceback inquiry, it is difficult to (1) pinpoint the cause – is it because the node

Figure 4-5: The maximum path length under the constraint of the traffic delivery ratio.
has moved away or it intentionally denied to collaborate? (2) Determine how far away the real attack source is from the broken link.

In spite of their shortcomings, these two schemes have their own strengths. In the marking-based scheme, a traceback process is an operation local to the victim. As long as the victim has received enough packets with authenticated marks and has space to store them, it can reconstruct the attack path anytime, even if the path has changed and its IDS takes long to respond. On the other hand, the logging-based scheme may trace back with a single packet since an edge in the path is verifiable by its two-ends— any disagreement between the two ends of an edge indicates at least one of them is compromised.

4.4 Online Traceback in MANETs

4.4.1 Scheme Overview

Our design is based on the principle of divide-and-conquer. To reduce the number of packets needed to reconstruct the entire path, we propose the ideas of path-fragmentation and fragment interweaving. With path-fragmentation, each packet during forwarding (probabilistically) divides its entire path into multiple fragments. The fragmentation information is stored in a few intermediate nodes. The fragments formed by multiple packets may be overlapped, building virtual interweaved links among en-route nodes. With these multiplicative and alternative trace routes, a broken link due to mobility could be compensated by other traceback routes and farther links can be reconstructed approaching to the real attack source. In addition, in the online traceback phase, the fragmentation information is gathered, each constructed link is verified, and finally a hotspot is identified.
For the illustration of the basic idea, let us consider Figure 4-6. A data packet $M_1$ is delivered through the path $A = (S, u_1, \ldots, u_5, V)$, where $u_1$ and $u_3$ are the markers in regard to $M_1$. Whenever an intermediate node decides to mark a packet, it must first log the existing mark in the packet before inscribing its own mark. Consequently, $u_1$ records the mark from $S$, $u_3$ records the mark from $u_1$, and $V$ records the mark from $u_3$. As a result, this path is divided into three fragments by $u_1$ and $u_3$ and a reverse virtual link is created pointing from one marker to its previous one.

Assume after $M_1$, another packet $M_2$ is sent through the same path $A$. The markers in regard to $M_2$ may be different, say $u_2$ and $u_5$, due to the probabilistic nature of the marking algorithm. Similarly, this path is divided into three fragments by $u_2$ and $u_5$ forming a different virtual link. As shown in Figure 4-6, the fragments of $M_1$ and $M_2$ interweave with one another. Two sets of reverse virtual paths are built along $A$ after $M_1$ and $M_2$ are sent through the path $A$.

The reverse virtual links help to localize the hotspot. In Figure 4-6, suppose $S$ is the attack source that has forged packet markings for $M_1$ and $M_2$. $u_1$ and $u_2$ recorded the false packet markings from $S$, respectively. If later $V$ initiates a traceback process can trace back to $u_1$ or $u_2$, and the markings stored in $u_1$ or $u_2$ indicates that they are spurious; then we may conclude that the attack source very likely resides nearby because these packet markings did not travel more than two or three hops generally (we will show how to achieve these goals shortly). Hence, we may
conclude that $u_2$, $u_1$ and $S$ are within the hotspot with a high confidence (recall our security assumption on authenticated links, which means $S$ must uses its own security credential to authenticate a packet to $u_1$). Compared to a logging scheme, our scheme does not have to physically trace back to $u_1$ along the forwarding path; the virtual link from $u_3$ can be used to identify $u_1$ and its neighbors as a hotspot.

To realize the above idea, we will construct three building blocks (BBs). During packet forwarding, every node employs a building block - "BB-I", a verifiable distance-based packet marking scheme to process every packet; which logging the abstract information of the packet into its traceback table. During a traceback process, BB-II, a multicast-based forwarding scheme, is employed to forward the traceback query. When a node finds that it was in the attack path, it reports to the victim its virtual links. Finally, the victim node calls upon the BB-III, a path-reconstruction algorithm, to identify the hotspot. Next we introduce these building blocks in greater details.

4.4.2 The Building Blocks (BBs)

**BB-I: Verifiable, Distance-based Packet Marking:** To provide path-fragmentation, it is critical to authenticate the packet markings so that only selected forwarding nodes are allowed to mark in a packet. It is also necessary to control the size of a fragment (i.e., the distance that a packet marking traverses) to increase traceability within each fragment.

A mark in our scheme has three fields: marker id, distance, and an authentication code. Note that unlike the IP traceback where only 16 bits are available for marking, in MANET we do not impose such a constraint. Let $H : \mathcal{K} \times \mathcal{D} \rightarrow \mathcal{R}$ be a keyed hash function, denoted as $H_k(\cdot)$, where $\mathcal{K}$ and $\mathcal{D}$ represents the domain of the hash key and the input data, respectively, and $\mathcal{R}$
denotes the domain of the hash result. For each packet $M$ destined at node $V$, an intermediate node $u_i$ calculates its marking probability $r^*$ as:

$$r^* = \frac{H_{K_{ui}}(M | u_i | V | d_{ui})}{|R|}$$  \hspace{1cm} (4-15)

where $K_{ui}$ is the symmetric key shared between $u_i$ and $V$, $d_{ui}$ is the distance between $u_i$ and the prior marker in packet $M$, and $|R|$ is the cardinality of the range of $H_k()$. $u_i$ then compares $r^*$ with a distance-based marking probability $p(d_{ui})$, as will be discussed shortly. If $r^* \geq p(d_{ui})$, $u_i$ is illegal to mark $M$ (i.e., for an intermediate node $u_i$, Equation (4-15) determines a unique $r^*$, which cannot be forged by $u_i$ itself and can be verified by the victim node $V$). Otherwise, $u_i$ replaces the existing mark with its own id $u_i$ and resets the distance field to 0. In addition, $H_{K_{ui}}(M | u_i | V | d_{ui})$ is written into the authentication code field.

This marking scheme not only authenticates the packet marking but also allows $V$ to check if the marker is legitimate to mark $M$. The security benefits are two folds: First, without knowing the key $K_{ui}$, an adversary cannot forge an authenticated mark. Second, even if the adversary compromises $u_i$ and gets the key, it still cannot arbitrarily select packets to mark. Note that in practice, we may set the range of $H_k()$ as $[0, 255]$, thus only one-byte cost is involved for authentication. The probability for an outsider attacker to inject a “legitimate” packet is only $p/256$.

No matter whether marking a packet or not, an intermediate node logs into its traceback table a hextuple containing such information as the packet digest, the prior marker, the distance to the prior marker, an authentication code, the forwarder (i.e., the immediate upstream node), and the destination. For MANETs, we will expect much lower traffic rate than that in high-speed Internet. As a result, in this work we expect the per-packet storage overhead being affordable to
mobile nodes and because we do not use Bloom filter mechanism to save storage. Indeed, for online traceback, we do not expect a long delay for launching a traceback, so old records will be overwritten.

**Determining the Marking Probability** The marking probability \( p \) in an ordinary PPM scheme is a fixed system parameter. From the security point of view, we prefer a packet marking not to travel too far. Consider an extreme case where each intermediate node checks if the packet mark inscribed by its one-hop neighbor. If the mark is inscribed by its neighbor then it will not mark the packet; otherwise, it overwrites the marking with its own id. In this case, a spurious mark added by the attack source will not propagate more than two hops (i.e., a two-hop hotspot). Nevertheless, from the traceability point of view, we prefer the packet markings to be delivered farther towards the destination to increase traceability under node mobility.

To determine the marking probability, we introduce the technique of adjusted probabilistic packet marking. In [118], the authors proposed to vary the marking probability from hop to hop according to the position of the marker in the path. As a result, the destination node can get the information of upstream nodes with fewer packets. In our scenario, we desire the majority of packet markings be overwritten by downstream nodes within two or three hops with a high probability so that when a spoofed mark is found in an intermediate node \( u_i \), a hotspot resides within three hops around \( u_i \) with a high probability. Moreover, as we will see later, the resulted virtual links also facilitate our online traceback to bypass broken links. On the other hand, we will preserve the long-distance traversal of packet markings at a low probability so that we may benefit more from marking in case that many attack packets do appear.

Specifically, our marking policy is as follows:
\[ p(d^*) = \begin{cases} 
    p_0 + (1 - p_0)(1 - e^{k(d^* - 1)}), & 1 < d^* < 4 \\
    \frac{1}{D^* - d^* + 1}, & 4 \leq d^* < D^* \\
    1, & d^* \geq D^* 
\end{cases} \] (4-16)

\(d^*\) denotes the distance from the prior marker. In Figure 4-6, for \(u_{11}, d^* = 2\) regarding \(M_1\). \(D^*\) refers to the upper bound of \(d^*\); when seeing \(D^*\) in the distance field, the downstream node is required to overwrite the marking. \(p_0\) is the marking probability when the node is two hops \((d^* = 2)\) away from the prior marker \((p(2) = p_0)\) (The one-hop prior node \((d^* = 1)\) is recorded by default). \(k\) is a tunable parameter. For example, Figure 4-7(a) shows a marking policy where \(p_0 = 0.35, k = 0.34,\) and \(D^* = 6\). As indicated in Figure 4-7(b), 70% packet markings are expected to traverse 2 to 3 hops, and the rest of 30% packet markings traverse the distance uniformly distributed among 4 to 6 hops. Hence, in this setting the average traverse distance is 3.25 hops.
BB-II: Multicast-based Forwarding of Traceback Queries: When a victim detects an attack, based either on a single packet or multiple ones, it initiates a traceback query and pushes the query toward the source in the reverse direction of the attack path. The challenge comes from the dynamics of the topology —one link disruption in the path could cause the failure of this unicast-based traceback.

To enhance the propagation of a traceback query, we apply the idea of virtual links aided multicast propagation. During a single-packet traceback or an aggregative traceback where a traceback query contains the message digests of multiple packets, an intermediate node $u_i$ first looks up its traceback table and identifies the hextuples that match the message digests. Then it puts the ids of the forwarders and the prior markers into the traceback query and broadcasts the query message to its one-hop neighbor. In addition, it replies to the victim along the reverse traceback path with all the matching hextuples. When a neighbor $u_j$ finds its id included in the received query message, it will process the query message in the same way as $u_i$ did; otherwise, it will ignore the message.
Unlike in a conventional logging scheme where a traceback fails when the immediate upstream node is out of reach, in our scheme, the traceback process can proceed when any node connected via a virtual link is in the neighborhood. As such, we expect higher traceability than in the logging scheme.

**BB-III: Path Reconstruction and Hotspot Detection:** The next issue is: after the victim has collected the reports from a set of nodes on the attack path, how will it reconstruct the attack path and identify the hotspot?

Every report should include one or multiple matching hextuples. For clarity, let us first consider the case the victim $V$ receives a report $R$ containing only one hextuple from a (claimed) node $u_i$ and the (claimed) prior marker is $u_j$. If the report is not authenticated by $u_i$, $V$ will discard it. Meanwhile, the victim may easily identify a hotspot through another traceback over this report packet.

Otherwise, $V$ first verifies the authentication code based on its pairwise key shared with the marker $u_j$ claimed in the report. If it is invalid, it will detect a hotspot starting from $u_i$. Here we cannot convict $u_j$ immediately because the id of $u_j$ may be spoofed by an attacker in the upstream of $u_i$ and $u_i$ may also be a malicious node that altered the authentication code. On the other hand, if an upstream node $u_x$ of $u_i$ reports the same invalid authentication code, $u_i$ will be considered innocent with regard to this packet $P$ and the hotspot is considered to start from $u_x$.

If the authentication code is valid but the marker $u_j$ is not authorized to mark based on the rule of verifiable packet marking, then $u_j$ is considered compromised; otherwise, if $u_j$ is authorized to mark $P$, $u_i$ will be considered a good one; the status of $u_j$ depends on additional reports. If $u_j$ is known as the farthest virtual link, then a hotspot is considered starting from $u_j$; otherwise, if any farther valid virtual link exists, $u_j$ is considered good with regard to $P$ and a hotspot is considered starting from the farthest virtual link. Figure 4-8 shows the above procedure.
This idea can be easily extended to the case of aggregative traceback or multiple single-packet tracebacks. In both cases, the victim receives multiple traceback reports regarding multiple packets. It will verify every one separately and crosscheck the correctness of each fragment.

**BB-IV: Node Reputation Evaluation:** The output of BB-III is a hotspot including multiple nodes. As we described in Section 4.2, once a hotspot is identified, we may rely on other online or offline analysis/detection measures (e.g., neighbor watching [30]) or human intelligence to pinpoint the attacker. Specifically, given that links are authenticated in our scheme, we can first identify the nodes in a hotspot. Then, we may aggregate the results of multiple traceback queries and keep track of the suspicious nodes in order to identify the attack source. For this purpose, we introduce a reputation system to handle the hotspot nodes. Many reputation systems have been applied to detect the misbehaving nodes in an ad hoc network [119][120]. In those systems, a node evaluates the reputation of a neighboring node by counting the frequency of its misbehavior. We introduce a similar idea to evaluate the suspiciousness of the nodes detected in the hotspot.

Although the reputation of a node can be represented by a number of distributions (e.g., Gaussian, Poisson, binomial, beta etc.), beta distribution is frequently adopted in the literature for its simplicity and its solid foundations on the statistics theory [117]. The Beta function is indexed by two parameters \( \alpha, \beta \) that denote the observations of the number of good behavior and bad behavior, respectively. Suppose a node is believed to have good behavior with probability \( x \) and bad behavior with probability \( 1-x \). According to Bayesian theorem, \( \alpha \) will be close to \( nx \) and
\( \beta \) will be close to \( n(1-x) \) through a series of \( n \) observations when \( n \) is very large. Initially, Beta function is set to \( \text{Beta}(1, 1) \), which is a uniform distribution on \([0,1]\). After a sequence of observations, the observed node has demonstrated \( S \) times good behavior and \( F \) times of bad behavior. Then the parameter \( \alpha \) is updated according to \( \alpha := \alpha + S \), and \( \beta \) is updated by \( \beta := \beta + F \). We expect \( \frac{\alpha}{\alpha + \beta} \approx x \) and \( E(\text{Beta}(\alpha, \beta)) = \frac{\alpha}{\alpha + \beta} \).

In this work, we maintain for each node \( u_i \) a reputation rating \( R_i \), which is defined as the expectation of the Beta function. At the beginning, \( R_i \) is set to \( E(\text{Beta}(1,1)) = \frac{1}{2} \). After each traceback procedure, for each node \( u_i \) in the hotspot the parameter \( \beta_i \) is incremented by one; for the node \( u_j \) which was previously in a hotspot but does not appear in the current hotspot, a small recovery factor \( \epsilon \in (0,1) \) is added to its parameter \( \alpha_j \). We summarize the \( \alpha / \beta \) update and the reputation rating as follows:

When \( u_i \) is in the hotspot,

\[
\begin{align*}
\alpha_i &:= \alpha_i \\
\beta_i &:= \beta_i + 1
\end{align*}
\]

(4-17)

When \( u_i \) is NOT in the hotspot but was previously in a hotspot

\[
\begin{align*}
\alpha_i &:= \alpha_i + \epsilon \\
\beta_i &:= \beta_i
\end{align*}
\]

(4-18)

and

\[
R_i = \frac{\alpha_i}{\alpha_i + \beta_i}
\]

(4-19)
4.5 Security Analysis

In this section, we outline the security threats and analyze the security properties of our traceback protocol. We do not consider outsider attacks as we assumed authenticated links. We do not consider other more subtle but generic attacks (e.g., the wormhole attack [21]), by assuming the countermeasures (e.g., packet leashes [21]) can be employed when needed (they are also required for securing routing protocols or others when these attacks are present).

We organize our discussion in the order of the three protocol phases —i.e., packet marking and logging, traceback queries forwarding, and traceback report; in each of the phase, we discuss the security threats and the security properties of our protocol.

4.5.1 Packet Marking and Logging Phase

**Mark Removal Attack:** A malicious intermediate node may intentionally remove the marks (i.e., set it to 0) from the forwarding packets —in order to reduce the connectivity between the upstream and downstream nodes or make some of the upstream nodes invisible to the downstream nodes.

Indeed, mark removal attack is a special case of mark replacement attack where an invalid id is inscribed. In either case, our scheme is capable of catching it because of our verifiable marking scheme where the authentication code must be correct and the marking probability must satisfy the condition. As such, packets with false marks will be detected. As long as we can trace to the next (downstream) marker of the malicious node, a hotspot will form nearby it.
4.5.2 Traceback Query Propagation Phase

**Query Tampering Attack:** A malicious intermediate node may alter the query message content, e.g., to hijack the traceback query by substituting the packet digest with an alternative digest such that the descending hops will trace a different route.

**Query Dropping Attack:** A compromised intermediate node in the attack path may silently drop all queries (black hole) or selectively drop some queries (gray hole).

Query tampering attack will not work in our scheme because every traceback query is authenticated by the victim. Although query dropping attacks can be detected by additional measures (e.g., neighbor monitoring [30]), our scheme does not rely on that. Indeed, the effect of query dropping is the same as that caused by node mobility. Our virtual link based traceback is designed to tolerate this path breaking issue and multi-round tracebacks also help avoid such malicious intermediate nodes.

4.5.3 Traceback Report Phase

**Report Tampering Attack:** A malicious intermediate node may alter the content of a report message from its upstream nodes.

**Report Dropping Attack:** Similar to the query dropping attack, the compromised intermediate node may also (selectively) drops the traceback reports.

A report tampering attack can be detected in our scheme because every report is authenticated by its source. When a report dropping attack occurs, the traceback process will stop at the attack node. Based on our policy of hotspot detection, unless this attack node reports a valid virtual link, it will be considered as the starting point of a hotspot. As such, even when it reports a valid virtual link itself, the hotspot is already multiple hops closer to the real attack source.
Moreover, due to the multicast-based traceback forwarding, a traceback operation may form a graph (not a tree because of the virtual links and changed topology); as a result, although this node may drop the reports going through it, there may exist other paths tracing closer to the real attack source. Of course, if additional measures (e.g., neighbor monitoring [30]) exist, the report dropping attack can be detected.

4.6 Performance Evaluation

In this section, we evaluate the performance of our online traceback scheme and use the simulation result to support our analysis. We use the hotspot size as the metric to indicate the effectiveness of our traceback protocol; for example, if we can trace to the immediate neighbor of the attack source, then the hotspot size is two. The smaller the hotspot size, the more accurate the traceback result, but the minimum hotspot size is two because an attacker can always accuse a good node.

In our simulation, we evaluate the impact of the attack packet rate, network mobility, and victim IDS response time. Specifically, we use the simulation results to show how the virtual links can help improve the efficacy of the online traceback.

Table 4-1: Simulation parameters.

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical link bandwidth</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Transmission range</td>
<td>250 m</td>
</tr>
<tr>
<td>Number of nodes</td>
<td>100</td>
</tr>
<tr>
<td>Territory</td>
<td>3000 m × 3000 m</td>
</tr>
</tbody>
</table>
4.6.1 Simulation Methodology

The GlomoSim 2.02 simulator [17] was used to simulate the proposed traceback protocol. For each parameter setting, we run the simulator for at least 40 times and take the average as the result. Each simulation run lasts 900 seconds in simulation time. Every node moves in random waypoint mobility pattern at a speed uniformly distributed between 0 and 10 m/s. In the network layer, we apply the AODV routing protocol. The major simulation parameters are listed in Table 4-1.

Our traceback protocol is implemented between the MAC layer and the network layer. We chose two nodes as the attack source and the victim node, respectively. The attack node sends packets to the victim node in a burst (e.g., 5 pkts/sec for 1 second) and waits for 10 seconds to try again as long as its distance to the victim is more than 3 hops (to reduce the risk of being easily recognized). An intermediate node follows the marking policy according to Figure 4-7. When the victim receives an attack packet, it waits for a response time to initiate a traceback query according to the traceback protocol. Then the victim node waits for a round trip time (1.5 sec) to receive the reports from the intermediate nodes. Finally, the victim node reconstructs the attack path and identifies the starting point of the hotspot based on the gathered traceback reports. We may then determine the hotspot size by calculating the distance between the starting point of the hotspot and the real attack source.

<table>
<thead>
<tr>
<th>MAC layer protocol</th>
<th>802.11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mobility model</td>
<td>Random-waypoint</td>
</tr>
</tbody>
</table>
4.6.2 Simulation Results

The Impact of Response Time and Packet Rate: Figure 4-9 shows that a larger response time causes a larger hotspot size, which indicates it is more difficult for the victim to traceback to the attack source. In addition, we observe that the higher packet rate results in a smaller hotspot size because more packets traversing through the attack path leaves more virtual links for the traceback query. In all the cases, the hotspot size is between 2 ~ 4.

![Figure 4-9: Hotspot size as a function of the IDS response time.](image)

The Impact of Mobility: Figure 4-10 indicates the average hotspot size enlarges as the network nodes have higher mobility. This is not surprising because the mobility reduces the traceability. However, even when the mobility of the network nodes vary from 5 to 20m/s, and the victim response time varies from 5 to 30 sec, the hotspot size on average is still bounded under 5 nodes. These results indicate that our scheme is relatively robust to node mobility.
The Realistic Mobility Model: Figure 4-11 demonstrates the simulation results based on a realistic mobility model. The mobility data is extracted from the dataset provided by CRAWDAD [48]. CRAWDAD is a well-known wireless network data resource for the research community, which archiving wireless data at Dartmouth. In this experiment, 35 laptops equipped with GPS receiver record its position while moving around. We converted the GPS position of the laptops
into the coordinate of simulator in order to generate the mobility traces which were then applied to the traceback simulations. We observe that the hotspot size under this mobility model is between $2 \sim 3$.

**The Benefit of Using Virtual Links:** To observe how virtual links help traceback, we also let the victim node try 9 traceback queries with the number of packet digests in each node ranging from 1 to 9. Then the victim node reconstructs the attack path separately for each of the query.

Figure 4-12 shows that with more packet digests embedded in a traceback query (or with more single-packet traceback attempts), the hotspot size decreases. Specifically, if only tracing a single packet, the hotspot size is 7.7, whereas after tracing 9 packets, the hotspot size is reduced to 3.5. This achievement is attributed to the interweaving virtual links, because more information toward the attack source can be gathered by the victim with more virtual links.

![Figure 4-12](image-url)

**Figure 4-12:** The benefit of virtual links. (The pkt# denotes the number of packet digests embedded in a traceback query).
**Comparison to the logging scheme:** Finally we compare our scheme to the logging scheme. It is fair for such comparison because in both schemes the authenticity of the reported links can be verified. We do not directly compare our scheme with a PPM scheme because of the weaker security (links cannot be verified) of PPM under insider attacks.

Figure 4-13 shows a comparison with the logging scheme. Here we can see on average our scheme outperforms a logging scheme at 2-3 hops and the larger the delay, the bigger the difference. Note that more packets in a logging scheme help little because their paths break at the same point.

![Figure 4-13: Comparison with logging-based traceback on hotspot size.](image)

### 4.7 Conclusions and Future Work

Traceback in MANETs is a challenging research issue because of node mobility and the lack of trust among mobile nodes. In this work we made the first effort to quantitatively analyze
the impacts of node mobility, attack packet rate, and intrusion response time on the traceability of PPM and logging schemes. Then we presented a new attack source traceback scheme. Our simulation study showed that our scheme could catch the attack source in a small hotspot and it outperforms the logging scheme in general.

Our future work will continue to improve the effectiveness of tracing attack sources in mobile environments. We will choose and evaluate other marking probability functions, and study the case of multiple attack sources. We also consider designing a traceback framework for MANETs that will work in a large spectrum of network settings by, for example, integrating the advantages of various schemes.
5.1 Introduction

Ad hoc wireless networks comprise a group of autonomous portable/mobile devices that generally rely on battery power. Due to the slow advance in the battery technology, the limited battery power duration remains to be the major constrain to the system lifetime, and energy continues to be a precious resource in ad hoc networks. As a result, the power efficiency has constantly being an important perspective in designing MANETs’ protocols.

In previous chapters, a security framework for MANETs was presented. The research discussion mainly focused on the security analysis and the network performance evaluation based on parameters, such as mobility, node density, location-awareness, and so on. In this chapter, the proposed protocols are evaluated with respect to energy efficiency. Specifically, the energy consumed by the original MANET protocol (e.g., AODV) is compared to the one integrated with the proposed security framework.

Section 5.2 presents the energy consumption model which consists of the computational and communicational power consumption. Based on this model, Section 5.3 evaluates the power efficiency of the LIP protocol. Section 0 presents the power efficiency of the proposed online traceback protocol. Finally Section 0 concludes this chapter.
5.2 Energy Consumption Model

5.2.1 Computational Energy Consumption

In light of the energy resource constraint in ad hoc networks, it is motivated to consider the computational energy costs of security function. In [112], the authors studied the computational energy consumed by security functions (e.g. hash, encryption/decryption etc.). According to their research, given a processor, the amount of energy consumed by a security function is determined by the processor clock frequency, the number of clocks needed to compute the security function, and the processor power consumption.

In [113], the authors estimated the computational power consumption based on the number of long number multiplication operations used in public key algorithms. Consider the following general formula for estimating the computational energy cost:

\[ E_X = N_{128 \cdot X} \cdot \frac{C_{128 \cdot X}}{f_Y} \cdot P_Y \]  \hspace{1cm} (5-1)

where the \( N_{128 \cdot X} \) refers to the number of 128-bits multiplication operations required for cryptographic operation \( X \), \( C_{128 \cdot X} \) denotes the number of clocks required for a 128-bits cryptographic operation, \( f_Y \) represents the frequency of the processor \( Y \), and \( P_Y \) represents the maximum power consumption by the processor \( Y \). For instance, MIPS R4000 has a clock frequency of 80 MHz, and its power consumption is 230 mW. The number of clock cycles required to compute a 128-bit multiplication is 40. Thus the energy consumption for a 128-bit multiply result is 115 nJ. Table 5-1 summarizes the specification.
Table 5-1: Computation time and energy consumption for 128-bit multiply result [112].

<table>
<thead>
<tr>
<th>Processor</th>
<th>Power (mW)</th>
<th>Clock rate (MHz)</th>
<th>Multiply result</th>
<th>( C_{128 \cdot x} )</th>
<th>Time (( \mu s ))</th>
<th>Energy (nJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIPS R4000</td>
<td>230</td>
<td>80</td>
<td>128</td>
<td>40</td>
<td>0.50</td>
<td>115</td>
</tr>
</tbody>
</table>

Table 5-2: Energy Consumption Estimates for digital signature.

<table>
<thead>
<tr>
<th>Processor</th>
<th>Clock (MHz)</th>
<th>RSA sign</th>
<th>RSA verify</th>
<th>DSA sign</th>
<th>DSA verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIPS R4000</td>
<td>230</td>
<td>16.7</td>
<td>0.81</td>
<td>9.9</td>
<td>20</td>
</tr>
</tbody>
</table>

Table 5-2 shows the estimated computational energy cost of the digital signature algorithms, RSA and DSA, respectively. It is calculated according to the multiply energy consumption result shown in Table 5-1. The reader may refer to [112] for more detail.

In contrary to the public key algorithms, the symmetric cryptographic operations (e.g. symmetric encryption/decryption and hashing functions) have significant lower computational energy consumption. In [112], the authors estimated the energy consumption of performing a 128-bit block encryption with a 128-bit key on a 32-bit microprocessor R4400; they conservatively estimated 400 clock cycles are used to perform this operation. Their estimation results indicate that the energy consumed by AES cryptographic algorithm is much lower than public key algorithms as well as the asymmetric cryptographic functions, such as RSA. Table 5-3 indicates the result of AES energy consumption estimate. For example, the energy consumed to encrypt a 1024-bit block using RSA is estimated 0.81 mJ while it only takes 0.0092 mJ to encrypt using AES.
Due to the limited resource in ad hoc networks, the authentication and integrity are generally provided by Hash-based Message Authentication Codes (HMAC). Given a message $x$ and a key $k$, HMAC can be computed by

$$HMAC(x) = MAC(k, x \mathbin\| p)$$

(5-2)

where $MAC$ is the hash function (e.g., SHA-1 or MD5) and $p$ is padding. For a message $x$ of length $m$, the hash function $MAC$ would be applied multiple times. In accordance with [112], the number of applying $h$ can be approximated by

$$\text{# of } MAC \text{ functions} = 1 + \left\lceil \frac{(m + 65)}{\text{block size}} \right\rceil$$

(5-3)

The block size for SHA-1 is 512 bits. Table 5-4 shows the computational energy cost on processor MIPS R4000 for SHA-1 hash algorithm. The energy cost of the HMAC is computed for a 1024-bit message.

Table 5-3: AES computational energy consumption estimate.

<table>
<thead>
<tr>
<th>Processor</th>
<th>AES Encrypt/Decrypt (mJ/bit)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIPS R4000</td>
<td>0.000009</td>
</tr>
</tbody>
</table>

Table 5-4: Energy consumption estimates.

<table>
<thead>
<tr>
<th>Processor</th>
<th>SHA-1 Energy (mJ/byte)</th>
<th>HMAC-SHA-1 Energy (mJ/1024-bit message)</th>
</tr>
</thead>
<tbody>
<tr>
<td>MIPS R4000</td>
<td>0.000058</td>
<td>0.0115</td>
</tr>
</tbody>
</table>
5.2.2 Communicational Energy Consumption

The communicational energy cost is generally a number of magnitudes larger than the computational energy cost [112]. Therefore, the energy consumption of the ad hoc network protocols are dominated by the communication cost. An energy consumption model of wireless network interface operating in ad hoc mode has been introduced in Section 2.3.1. This communicational energy consumption analysis model will be used in our simulation.

5.3 LIP – Lightweight hop-by-hop authentication protocol

The analysis of power efficiency of the proposed LIP protocol (See Chapter 3) consists of two parts: 1) the computational energy analysis, which evaluates the computational energy consumption based on the complexity of the cryptography functions that have been used in the proposed protocol. 2) The communicational energy analysis, which evaluates the communicational energy consumption based on the control messages exchanged between the network nodes according to the protocol.

5.3.1 Computational energy analysis

In the computational energy estimation, the simulation only considers the energy after the start of protocol operation. The experiment does not consider the pre-deployment overhead, such as the key distribution, key generations, the network membership initialization, and so on. That one-time initialization may be performed offline and are less relative to the energy efficiency for each individual node. The following analysis is based on the following assumptions:

- The packet index $i$ is 16 bits.
The size of data packets $P(i)$ are 512 bytes.

Cluster keys $K(i)$ are 10 bytes.

The coordinator and velocity information are 8 bytes, respectively.

The message authentication code $MAC$ is 10 bytes.

**Per-packet authentication:** As shown in Equation (3-1), the sender $u$ generates a $MAC$ for each data packet, and the receiver $v$ calculates the same $MAC$ to verify the sender. Thus, each node calculates a $MAC$ for the message of size $(i + P(i))$, which is 4128 bits. According to Table 5-4, it consumes about 0.046 mJ.

**Random neighborhood verification:** Equations (3-3) and (3-4) show that whenever a receiver $v$ receives a data packet from sender $u$, it verifies the neighborhood of the sender by sending the request message back to $u$ with a probability $p_c$. In this process, node $v$ calculates a $MAC$ for the attached $MAC$ of data packet, which is of size 80 bits. On the other hand, node $u$ calculates a $MAC$ for the response which includes a packet index of size 16 bits and a $FLAG$ of one bit. The following table summarizes the energy consumption of the two nodes:

**Table 5-5:** The estimation of the computational power of packet authentication.

<table>
<thead>
<tr>
<th>Equation (3-1): $M : i, P(i), K(i+1) \oplus MAC(K(i), i | P(i))$</th>
<th>Sender $u$ Energy (mJ)</th>
<th>Receiver $v$ Energy (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MAC$</td>
<td>0.046</td>
<td>0.046</td>
</tr>
<tr>
<td>Encryption</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Signature</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total (mJ)</td>
<td>0.24</td>
<td>0.24</td>
</tr>
</tbody>
</table>
Location-aware verification: An alternative way to verify the neighborship between network nodes is to exchange location information, including the current coordinate and velocity, and predict the duration of the neighborship. The procedure incorporates cluster key exchange and message authentication. As shown in Equations (3-5) and (3-6), both sender and receiver calculate an encryption for the commitment of cluster key and a MAC for the message. The commitment of cluster key is of size 80 bits, and the authenticated message is of size 224 bits (2 bytes of packet index, 10 bytes of cluster key, and 16 bytes of location information). According to Table 5-3 and Table 5-4, the computational energy consumption is estimated as follows:

Table 5-6: The estimation of the computational power of neighborship verification.

<table>
<thead>
<tr>
<th></th>
<th>Sender $v$ Energy (mJ)</th>
<th>Receiver $u$ Energy (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MAC$</td>
<td>0.0009</td>
<td>0.0002</td>
</tr>
<tr>
<td>Encryption</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Signature</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total (mJ)</td>
<td>0.0009</td>
<td>0.0002</td>
</tr>
</tbody>
</table>

Equation (3-3): $v \xrightarrow{p} u : i, MAC(K_w, MAC(K(i), P(i)))$

Equation (3-4): $u \rightarrow v : i, MAC(K_v, i \parallel FLAG), FLAG$

<table>
<thead>
<tr>
<th></th>
<th>Sender $u$ Energy (mJ)</th>
<th>Receiver $v$ Energy (mJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$MAC$</td>
<td>0.005</td>
<td>0.005</td>
</tr>
<tr>
<td>Encryption/Decryption</td>
<td>0.00144</td>
<td>0.00144</td>
</tr>
<tr>
<td>Signature</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Total (mJ)</td>
<td>0.00644</td>
<td>0.00644</td>
</tr>
</tbody>
</table>
5.3.2 Communicational energy analysis

This section estimates the communicational energy consumption of LIP. Analogous to the computational energy analysis, the energy consumption evaluation is introduced for each protocol message; then the overall energy consumption is evaluated via simulation. Recall the power consumption model introduced in Section 2.3.1, as shown in Equation (5-4).

$$\text{Energy} = m \times \text{size} + b.$$  (5-4)

For each control message in LIP, the corresponding parameters are: $m$, $\text{size}$, and $b$. Notice that the parameters are determined according to its communication model; for example, the parameters for point-to-point communication are different from that of the broadcast communication.

**Per-packet authentication:** As shown in Equation (3-1), the sender $u$ generates a $MAC$ for each packet. Table 5-7 depicts the communicational power consumed for each data packet delivery.

Table 5-7: The estimation of the communicational power of packet authentication.

<table>
<thead>
<tr>
<th>Equation (3-1): $M : i, P(i), K(i + 1) \oplus MAC(K(i),i \parallel P(i))$</th>
<th>Sender $u$</th>
<th>Receiver $v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$ ( $\mu W \cdot \text{sec/byte}$ )</td>
<td>1.9</td>
<td>0.5</td>
</tr>
<tr>
<td>$b$ ( $\mu W \cdot \text{sec}$ )</td>
<td>454</td>
<td>356</td>
</tr>
<tr>
<td>$\text{size}$ (bytes)</td>
<td>524</td>
<td>524</td>
</tr>
<tr>
<td>Energy (mJ)</td>
<td>1.4496</td>
<td>0.618</td>
</tr>
</tbody>
</table>
**Random neighborship verification:** On receiving a data packet from node $u$, node $v$ unicasts a challenge to node $u$ with a probability $p_c$. The power consumed in this communication is shown in Table 5-8.

Table 5-8: The estimation of the communicational power of neighborship verification.

<table>
<thead>
<tr>
<th>Equation (3-3): $v \xrightarrow{p_c} u : i, MAC(K_w, MAC(K(i), P(i)))$</th>
<th>Sender $v$</th>
<th>Receiver $u$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$ ($\mu W \cdot sec/byte$)</td>
<td>1.9</td>
<td>0.5</td>
</tr>
<tr>
<td>$b$ ($\mu W \cdot sec$)</td>
<td>454</td>
<td>356</td>
</tr>
<tr>
<td>size (bytes)</td>
<td>12</td>
<td>12</td>
</tr>
<tr>
<td>Energy (mJ)</td>
<td>0.4768</td>
<td>0.362</td>
</tr>
</tbody>
</table>

Equation (3-4) shows the response from node $u$.

<table>
<thead>
<tr>
<th>Equation (3-4): $u \rightarrow v : i, MAC(K_w, i | FLAG), FLAG$</th>
<th>Sender $u$</th>
<th>Receiver $v$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$m$ ($\mu W \cdot sec/byte$)</td>
<td>1.9</td>
<td>0.5</td>
</tr>
<tr>
<td>$b$ ($\mu W \cdot sec$)</td>
<td>454</td>
<td>356</td>
</tr>
<tr>
<td>size (bytes)</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>Energy (mJ)</td>
<td>0.48</td>
<td>0.363</td>
</tr>
</tbody>
</table>

**Location-aware verification:** When both of the neighboring nodes are location-aware, they not only exchange cluster keys for neighborship verification but also exchange the position and speed information; hence, they will be able to more precisely predict their neighborship duration. The power estimation for this communication is shown in Table 5-9.
Section 5.3.1 and 5.3.2 gave a qualitative analysis for each control and traffic message in LIP. In this section, the quantitative results are presented via simulation. The overall energy consumption varies with the network settings. The experiments evaluate the impact of node mobility, node density, location awareness, and traffic load. Each of the impacts is examined under two routing protocols: AODV and DSR.

We use the same simulator (GloMoSim 2.02 [17]) as the one which we used to estimate the performance of LIP and online traceback in previous chapters. In addition, we developed and

Table 5-9: The estimation of the communicational power of neighborship verification.

| Equation (3-5): $u \rightarrow v : LP_u,0,\{K_u(0)\}_K, MAC(K_u,0 || K_u(0) || LP_u)$ |
|---------------------------------------------------------------|
| Sender $u$ | Receiver $v$ |
| $m$ (μW·sec/byte) | 1.9 | 0.5 |
| $b$ (μW·sec) | 454 | 356 |
| size (bytes) | 38 | 38 |
| Energy (mJ) | 0.5262 | 0.375 |

| Equation (3-6): $v \rightarrow u : LP_v,i,\{K_v(i)\}_K, MAC(K_v,i || K_v(i) || LP_v)$ |
|---------------------------------------------------------------|
| Sender $v$ | Receiver $u$ |
| $m$ (μW·sec/byte) | 1.9 | 0.5 |
| $b$ (μW·sec) | 454 | 356 |
| size (bytes) | 38 | 38 |
| Energy (mJ) | 0.5262 | 0.375 |

5.3.3 Simulation Results

Section 5.3.1 and 5.3.2 gave a qualitative analysis for each control and traffic message in LIP. In this section, the quantitative results are presented via simulation. The overall energy consumption varies with the network settings. The experiments evaluate the impact of node mobility, node density, location awareness, and traffic load. Each of the impacts is examined under two routing protocols: AODV and DSR.
integrated the power estimation model into the original simulation code in order to measure the computational and communicational power dissipation, respectively. We present the simulation set up in Table **5-10**. The simulation settings are basically the same to those that have been used in the performance evaluation of LIP and online traceback. These parameters are the typical settings that have been widely adopted in the literature. For instance, the radio settings result in the effective radio range of 250m, which is consistent to the real radio experimental results [123].

**Table 5-10: Simulation parameters.**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Physical link bandwidth</td>
<td>2 Mbps</td>
</tr>
<tr>
<td>Radio frequency</td>
<td>$2 \times 10^9$ Hz</td>
</tr>
<tr>
<td>Radio transmission power</td>
<td>15 dBm</td>
</tr>
<tr>
<td>Radio transmission sensitivity</td>
<td>-91 dBm</td>
</tr>
<tr>
<td>Radio transmission threshold</td>
<td>-81 dBm</td>
</tr>
<tr>
<td>Mobility model</td>
<td>Waypoint</td>
</tr>
<tr>
<td>Location information size</td>
<td>8 bytes</td>
</tr>
<tr>
<td>Velocity information size</td>
<td>8 bytes</td>
</tr>
<tr>
<td>HMAC key size (including a key id)</td>
<td>10 bytes</td>
</tr>
</tbody>
</table>

**The Impact of node mobility:** Figure 5-1 shows the estimation of the computational energy consumption normalized to each node based on the simulation results. The figure indicates that the power consumption of AODV routing protocol increases significantly than DSR routing
protocol. This is because AODV maintains routing tables where every node periodically broadcasts HELLO messages to neighbors to update the current network topology. Therefore, AODV is more susceptible to the mobility.

Figure 5-1: The impact of the node mobility on the computational power consumption.

Figure 5-2: The impact of the node mobility on the communicational power consumption.
Figure 5-2 shows the simulation results for the estimation of the communicational power consumption. Analogous to Figure 5-1, this figure also suggests AODV has higher communication overhead when the network mobility increases.

**The Impact of node density:**

![Graph showing the impact of node density on communicational power consumption](image)

Figure 5-3: The impact of the node mobility on the computational power consumption.

In the experiment of evaluating the node density impact, one hundred of nodes are uniformly distributed in a territory varying from \((500\text{m} \times 500\text{m})\) to \((2000\text{m} \times 2000\text{m})\); the smaller territory reflects the higher density of network nodes. All of the other experiment settings are the same to the simulations conducted in Chapter 3. Figure 5-3 demonstrates the estimation of the computational and communicational power consumption per node. The simulation results suggest that AODV protocol dissipates more power when the network is denser while DSR behaves in the opposite way. The reason for AODV is that in a denser network, each node has more neighbors and more nodes reside within the fixed transmission range; hence, every node receives more HELLO messages from neighbors and the probability of packet collision and retransmission is higher. On the other hand, DSR has lower power consumption in a smaller territory and denser
network because the network routes are shorter (less hops) on average and each route links are less likely to break up given the fixed transmission range and the shorter distance between the network nodes.

Figure 5-4: The impact of the location awareness of the nodes.
The Impact of location awareness: Figure 5-4 illustrates that the computational power of AODV and DSR increases with the percentage of location-aware nodes; while the communicational power overhead of AODV decreases with the percentage of location-aware nodes (the case of DSR is not significant in this simulation). This is because of the extra location verification which increases the computational overhead for AODV while the more precise estimation of the neighborship between network nodes reduces the number of periodical neighborship verification messages (HELLO), and thus reduces the communicational power. According to the experimental results, we may conclude that the AODV routing protocol is more sensitive to the deployment of LIP in terms of power efficiency because AODV requires periodical broadcast for topology maintenance. On the other hand, DSR does not conduct periodical routing table maintenance; therefore, the power overhead increases with the percentage of the location-aware nodes when location information are embedded in the neighborship verification messages.

![Figure 5-5: The communicational overhead of LIP versus LHAP.](image-url)
Figure 5-5 indicates that the proposed protocol LIP imposes less power dissipation than the existing protocol LHAP [43] in the literature. Instead of periodically sending out HELLO messages for neighborship verification, LIP adopts a more flexible approach where the data packet receivers randomly challenge the senders and every node roughly predicts the neighborship duration of its neighbor nodes; therefore, LIP protocol successfully reduced the amount of communicational power dissipation.

5.4 Online traceback for MANETs

In the hop-by-hop authentication protocol, a static amount of energy is consumed for each data packet delivery. Power efficiency is determined by the network configuration (e.g., the mobility, the node density, and the prevalence of location-aware nodes). Contrarily, in online traceback protocol, the victim determines whether to launch a traceback procedure when a malicious traffic is detected. Obviously, the more frequently the victim attempts online traceback the more energy is consumed by the traceback protocol. Chapter 4 only considers how effective the online traceback protocol can trace to the attack source without considering the cost incurred by the online traceback protocol. This section will evaluate the online traceback protocol from the perspective of power efficiency. In other words, a traceback attempt is no longer free; the victim should take into account both of the traceability and the power efficiency.

This section first analyzes the power overhead for each traceback query. Then an experimental scenario is considered where given a packet rate, the victim node determines the frequency to launch traceback query for the received malicious packets. Specifically, on one hand, the victim may launch a traceback query for each individual received packet; on the other hand, the victim may launch an aggregative traceback for a group of received packets. Intuitively, the more frequent traceback queries consume more energy. The goal is to determine a traceback
frequency at which the system saves more energy while preserving the protocol traceability. The issue is addressed by analyzing the simulation results.

### 5.4.1 Computational energy analysis

The proposed online traceback protocol is designed lightweight; i.e., most of the cryptographic operations committed in this protocol are hash computations. This section evaluates the computational and communicational energy consumption of the online traceback protocol based on its three protocol phases: packet marking and logging phase, traceback query forwarding phase, and traceback report phase. Similar to Section 5.3, the pre-deployment energy overhead is not considered. The analysis is based on the following system setup:

- The size of node ID (e.g., $u_i$ and $V$) is 4 bytes.
- The packet size of $M$ is 512 bytes.
- The size of the output of $HMAC$ is 16 bytes (128 bits).
- The size of random number (e.g., $R$ and $r^*$) is 4 bytes.
- The size of pairwise key $K_{uv}$ is 10 bytes.
- Traceback request $RREQ$ is of size 64 bytes.
- Traceback report $RREP$ is of size 66 bytes.

**Packet marking and logging phase:** As described in Section 4.4.2, each intermediate node decides whether or not to mark the forwarding packet $M$ based on calculating the pseudo random number $r^*$. The energy consumed for this calculation is as follows:
The intermediate node calculates a $MAC$ for the message $M || ui || V || d_u$ where $M$ is the forwarding packet (512 bytes), $ui$ and $V$ are the identities of the intermediate node and the victim node, respectively (4 bytes), and $d_u$ is the distance to the previous marking node (2 bytes); therefore, the message is length size 522 bytes. According to Table 5-4, it consumes about 0.03 mJ.

### 5.4.2 Communicational energy analysis

The communication overhead of the online traceback protocol consists of the multicast traceback query messages (TBREQ) and the unicast traceback report (TBREP).

**Traceback query forwarding phase:** In the traceback query phase, the victim node $V$ launches a traceback query message TBREQ toward the attack source in a multicast manner. The format of the query message is TBREQ = $[seq, IDs, V, h(M)]$ where $seq$ is the sequence number (4 bytes), $IDs$ is the list of identity of $n$ ($n \leq 10$) multicast hops (4 bytes each), $V$ is the identity of victim node $V$ (4 bytes), and $h(M)$ is the packet marking extracted from the received data packet (16 bytes).
bytes). Therefore, the size of the message TBREQ is 64 bytes. Based on the power consumption model in Section 5.3.2, the communicational power consumption is summarized as follows:

\[
\text{TBREQ} = [\text{seq}, \text{IDs}, V, h(M)]
\]

<table>
<thead>
<tr>
<th></th>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m \ (\mu\text{W}\cdot\text{sec}/\text{byte}) )</td>
<td>1.9</td>
<td>0.5</td>
</tr>
<tr>
<td>( b \ (\mu\text{W}\cdot\text{sec}) )</td>
<td>454</td>
<td>356</td>
</tr>
<tr>
<td>size (bytes)</td>
<td>64</td>
<td>64</td>
</tr>
<tr>
<td>Energy (mJ)</td>
<td>0.576</td>
<td>0.388</td>
</tr>
</tbody>
</table>

**Traceback report forwarding phase:** In the traceback report phase, each intermediate node \( u_i \) on the route which received the query TBREQ unicasts a traceback report TBREP backward to the victim node \( V \). The format of the traceback report is \( \text{TBREP} = [id, \text{seq}, \text{hops}, \text{IDs}, h(M)] \), where

- \( id \) is the identity of the reporting intermediate node \( u_i \) (4 bytes),
- \( \text{seq} \) is the sequence number of the corresponding query TBREQ (4 bytes),
- \( \text{hops} \) is the distance in hop from the victim node \( V \) (2 bytes),
- \( \text{IDs} \) is the list of identities of each hops along the route from \( u_i \) to \( V \) (4 bytes each), and
- \( h(M) \) is the packet digest copied from the corresponding TBREQ (16 bytes). Hence, the total size of each traceback report TBREP is also 66 bytes. Similarly, the communicational power consumption is summarized in the following table:

\[
\text{TBREP} = [id, \text{seq}, \text{IDs}, h(M)]
\]

<table>
<thead>
<tr>
<th></th>
<th>Sender</th>
<th>Receiver</th>
</tr>
</thead>
<tbody>
<tr>
<td>( m \ (\mu\text{W}\cdot\text{sec}/\text{byte}) )</td>
<td>1.9</td>
<td>0.5</td>
</tr>
<tr>
<td>( b \ (\mu\text{W}\cdot\text{sec}) )</td>
<td>454</td>
<td>356</td>
</tr>
</tbody>
</table>
This section evaluates the power consumption of the traceback protocol over three variants: the traceback query frequency, the packet rate of the attack source, and the node density of the network. Similarly, we use GloMoSim 2.02 [17] as our simulator, and the simulation settings are the same to those of the performance evaluation in Chapter 4. The power evaluation model is developed and incorporated into the original simulation code to evaluate the power consumption under each traceback protocol phases.

**The impact of traceback query frequency**

In this experiment, the attack node sends malicious packets to the victim node at a constant rate (1pkt/sec). The victim node determines to launch a traceback query for each malicious packet or launch an aggregative traceback request for a collection of received malicious packets. Obviously, the more traceback queries launched the more energy consumed.

(a) The traceback frequency versus the power consumption.
Figure 5-6 (a) shows that the victim node consumes less energy if it collects a number of malicious packets and launches an aggregative traceback query for them; on the other hand, Figure 5-6 (b) shows that launching the traceback queries in less frequency still preserves the overall traceability. Therefore, it would be advantageous to the victim node to conduct the aggregative traceback.

**The impact of packet rate**

In this experiment, the number of digest embedded to each traceback query is fixed to five digests; that is, the victim node launches a traceback query for every five received malicious packets. Under this condition, the experiment varies the packet rate at the attack source from 1 packet per five seconds to four packets per second.

Figure 5-7 shows the averaged power consumption per node and the traceability (hotspot size) of the traceback protocol varying with the attack source packet rate. Figure 5-7 (a) shows that the overall traceback power consumption increases with the attack source packet rate. This is because of the fix number of embedded digests in the traceback query by the victim node; the
higher packet rate results in more frequent traceback queries and thus incurs higher energy overhead.

The Figure 5-7 (b) depicts that when the packet rate increases over 1 packet/sec, the traceability starts to decrease (expanded hotspot size). That is because the over congested route increases the probability of packet collision and the packet lost due to buffer overflow of the intermediate nodes. This experiment result suggests that the victim node should cap the frequency of launching the traceback query for both traceability and power efficiency; otherwise, it could be utilized by the adversary to launch a denial of service (DoS) attack.

(a) The packet rate versus the power consumption.
The impact of node density

The node density will affect how many neighboring nodes will overhear the traceback query and report, and how easily the traceback query may propagate toward the attack source. Figure 5-8 illustrates that the power consumption and the hotspot size decrease with the node density but they rise again when the number of nodes increases over 140. The reason for this phenomenon is analogous to that of the over high packet rate; the over-crowded network nodes may increase the possibility of packet collisions between the neighboring nodes, and the possibility of dropping the forwarding packets in the intermediate nodes due to buffer overflow.

Figure 5-7: The power consumption and the hotspot size versus the packet rate from the attack source.
The node density versus the power consumption.

The node density versus the hotspot size.

Figure 5-8: The power consumption and the hotspot size versus the node density.
Figure 5-9 compares the communicational power overhead of our online traceback scheme against the previous work in the literature [58], which floods the traceback request throughout the entire network. In this experiment, a hundred of nodes are uniformly distributed in various area of territory ranging from (500m × 500m) to (2000m × 2000m). As shown in this figure, our scheme consumes much less communicational power in contrary to the flooding approach. This is one of the significant advantages of our scheme over the previous work. We trade part of the traceability for higher performance and power efficiency. In reality, this is a more feasible approach to deploy the online traceback protocol in MANETs.

5.5 Conclusion

In this chapter, an energy consumption model is introduced and applied to estimate power consumption of the proposed security frameworks. Based on the energy consumption model, extensive simulations were conducted to evaluation the power efficiency of the LIP and online traceback, under various system setups. In addition to the conventional performance evaluation
metrics of the security protocols, the simulation results provide an insight from the perspective of power efficiency. It would be a good reference for the deployment and the operation of the security framework in a real mobile ad hoc network.
Chapter 6

Concluding Remarks and Future Work

This chapter summarizes our research contributions, and further discusses several directions that one can take to advance the scope of this work.

6.1 Contributions of this Thesis

Unlike Internet, MANETs is a system of autonomous mobile nodes, a network of lacking trustworthy intermediate gateways, a network upon open media accessible to anonymous nodes, and an isolated environment where each node defends itself. Consequently, the security challenges are more severe, and a wide variety of attacks could exploit its weaknesses. This research classifies the attacker model into two categories: outsider attack and insider attack. The outsider attackers do not possess proper system commitment and cannot legally communicate with the network nodes. However, they may exhaust the network resources by launching DoS/DDoS attacks. The insider attackers may generate legal data packets, attack other network nodes, and hide themselves by spoofing packet headers. The proposed MANETs security framework counteracts these security threats while preserving power efficiency with small overhead. In summary, this research aims at the following security and power efficiency challenges:

- How to restrict network access to non-authorized nodes while imposing diminutive overhead to the original network?
- How to localize the attack source without flooding the queries throughout the mobile ad hoc network?
• What is the relationship between the traceability and the path length?
• What is the impact on network performance by introducing the network access control and the traceback scheme?
• What is the impact on power efficiency by introducing the network access control and the traceback scheme?

Our study has contributed to a number of research fields: hop-by-hop authentication-based network access control, online attack source traceback for MANETs, and power efficiency evaluation. The following summarizes our research in the aforementioned directions.

**Hop-by-hop Authentication-based Network Access Control** – In Chapter 3, we proposed an efficient, scalable, and general-purpose network access control protocol which prevented packet injection attacks in ad hoc networks. Our solution is designed based on the lightweight localized broadcast authentication mechanisms in which only a hash function is utilized to generate the per packet authentication code. By verifying neighborship periodically, only authenticated neighbors are authorized to receive data packets. Our scheme provides a much stronger network access control capability than a network-wide key based scheme; it also avoids computing digital signatures and thus imposes small computational overhead. In addition, our scheme can be integrated seamlessly with secure routing protocols to provide strong security framework for a MANET.

**Online Attack Source Traceback for MANETs** – In Chapter 4, we made the first effort to quantitatively analyze the pros and cons of the marking-based and logging-based traceback approaches. In particular, we derived the relationship between the network settings (e.g., the mobility, the path length and the victim response time) and the traceability of these approaches. According to our study results, we concluded that: (i) The marking-based approach is vulnerable to low packet rate attacks. (ii) The logging-based approach performs poorly on the protracted victim response time. (iii) The traceability of both of these approaches decreases with node
mobility. We also proposed a novel online traceback for MANETs. The proposed scheme is based on the principle of divide and conquer where the attack path is divided into multiple interweaving fragments. Our solution also included three features: verifiable and distance-based packet marking, multicast-based traceback query forwarding, and hotspot-based attacker detection. The first feature prevents the non-trustworthy intermediate nodes in MANETs; the second feature strengthens logging-based traceability via bypassing the broken links; the third feature minimizes the suspicious area (hotspot) by analyzing the traceback reports.

**Power Efficiency Evaluation** – In Chapter 5, we analyzed the power efficiency of the proposed security schemes using a power consumption model; the analysis considered both of the computational power consumption and the communicational power consumption. Specifically, we studied the relationship between the network settings (e.g., the node mobility, the node density, and the percentage of location-aware nodes) and the power efficiency for security schemes proposed in Chapter 3 and 4.

In summary, we believe that the proposed security mechanisms are effective, lightweight, secure, and power efficient. It can be integrated seamlessly with other security mechanisms, such as the secure routing protocols and neighbor monitoring protocols, to provide an integral security framework for MANETs.

### 6.2 Future Research

Our research can be extended in the following directions:

- At present, the neighborship verification in the network access control protocol is based on the challenge-and-response mechanism which incurs additional communication overhead. Although the GPS information is utilized to increase the precision of the neighborship prediction and reduce the frequency of
neighborship verification, it is not directly involved in the neighborhood authentication. We believe integrating the geographic information into the authentication/verification mechanism or the cluster keys could exempt the need of issuing challenge messages to neighbors, and thus further reduce the communication overhead of the protocol.

- The proposed online traceback protocol only considers the single attack source model. In reality, the adversary may launch attack from multiple attack sources (e.g., DDoS attack). The multiple attack sources is a more complicated and difficult issue to be address than the single attack source traceback. Applying appropriate coding technique to packet marking, such as the algebraic packet marking, may be useful to resolve this problem.

- The present online traceback protocol can only localize region (the hotspot) where the attack source reside in. To pinpoint the real attack source in the online traceback problem, a neighbor monitoring scheme is needed to complement the accuracy of the online traceback scheme. A controlled query flooding in the hotspot may be a feasible solution.
Bibliography


[48] CRAWDAD metadata: Dartmouth/outdoor (v.2006-11-06),


https://www.trustedcomputinggroup.org/groups/tpm/


VITA

Hung-Yuan Hsu is a Ph.D. candidate of the Computer Science and Engineering Department at the Pennsylvania State University. He received his M.S. in Computer Science and Information Engineering from National Taiwan University, Taipei, in 1999 and his B.S. in Computer and Information Science from National Chiao Tung University, Hsinchu, Taiwan, in 1997. His research interests include network access control, online traceback, and power efficiency in mobile Ad Hoc Networks.