1-1-2012

Preventing Collaborative Wormhole Attacks In AODV-based Mobile Ad-Hoc Networks

Vincent Koo
Ryerson University

PREVENTING COLLABORATIVE WORMHOLE ATTACKS IN AODV-BASED MOBILE AD-HOC NETWORKS

by

Vincent Koo

B.Sc., Ryerson University, Toronto, Ontario, Canada, 2010

A Thesis

Presented to Ryerson University

in partial fulfilment of the

requirements for the degree of

Master of Science

in the Program of Computer Science

Toronto, Ontario, Canada, 2012

AUTHOR'S DECLARATION

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I authorize Ryerson University to lend this thesis to other institutions or individuals for the purpose of scholarly research.

I further authorize Ryerson University to reproduce this thesis by photocopying or by other means, in total or in part, at the request of other institutions or individuals for the purpose of scholarly research.

I understand that my thesis may be made electronically available to the public.

Abstract

Preventing Collaborative Wormhole Attacks on AODV-Based Mobile Ad Hoc Networks

Master of Science
Computer Science
Ryerson University

Due to recent advances in wireless communication technologies, mobile ad hoc networks (MANETs) have become the networks of choice for use in various applications. Unfortunately, this advantage comes with serious security concerns, particularly from a wireless channel prospective, where MANETs may be vulnerable to collaborative wormhole attacks, in particular, packet dropping and message tampering attacks. Recently, two secured routing protocols against these types of attacks (the so-called Highly Secured Approach Method or HSAM and the so-called Securing AODV against Wormhole Attacks in Emergency MANET Multimedia Communications or AODV-WADR-AES) were proposed. These schemes are investigated in-depth to identify their weaknesses and a novel secured routing scheme (so-called Timed and Secure Monitoring Implementation against Wormhole Attacks in MANETs or TSMI) is proposed, with the goal to mitigate collaborative wormhole attacks in MANETs while maintaining the performance levels of both the HSAM and AODV-WADR-AES schemes. Simulation results are presented to validate the stated goal using various performance metrics.

Acknowledgement

I would like to first thank my supervisor, Dr. Isaac Woungang, and my co-supervisor, Dr. Sanjay Kumar Dhurandher, for their support, patience and time throughout my graduate studies. They have truly made my journey as a graduate student an enjoyable and fruitful one.

I would like to acknowledge the Department of Computer Science and the School of Graduate Studies at Ryerson University for their support in terms of financial aid and work experience as a graduate assistant. I would like to especially thank Deborah McKay, Lucia Flaim and all front office staff for their dedication and patience (with me especially).

Thank you to my defence committee for taking the time and effort to review my work and provide me with their insightful comments.

My fellow graduate peers deserve many thanks as we travelled this journey together throughout our ups and downs. I am going to miss complaining about stuff to you guys and going out for drinks. I assure all of you that I will now go out and find a permanent desk to sit at! Dheeraj "The Kid" Peddi, Sweeney "Cartoon" Luis, Subir "Hero" Biswas and Rusho "Barca" Islam, let us never forget our bi-weekly Fridays! Long live the chipotle wrap!

I would like to thank my family for their timeless wisdom, patience and support. Without them I would not be here today. This thesis and degree is as much theirs as it is mine. My brother for his invaluable advice and support. Bro, you are always there when I am stuck. Thanks for helping me out when I really needed you. My mom and dad for their nurturing nature and unconditional love. You guys stood by me when I was at my weakest and never gave up on me when I needed support most. My grandfather and grandmother for their love and support. I would also like to express my appreciation to my girlfriend, Anna Yeung, for her support, stress relieving smile, crazy adorable-ness and other kissy-mushy stuffies (pig farm night!).

Contents

List of Figures

List of Tables

List of Algorithms

List of Abbreviations

ACK	Acknowledgement
AES	Advanced Encryption Standard
AODV	Ad Hoc On-Demand Vector
AODV-WADR	AODV-Wormhole Attack Detection Reaction
AODV-WADR-AES	AODV-WADR Advanced Encryption Standard
AODV-WADR-TDES	AODV-WADR Triple Data Encryption Standard
ATT	Actual Traversal Time
ATT_WADR	Actual Traversal Time Wormhole Attack Detection Reaction
Blacklist_wa	Blacklist Wormhole Attack
CBR	Constant Bit Rate
Cmiss	Counter Packet Missed
Cpkt	Counter Packet Received
DES	Data Encryption Standard
DH	Diffie-Hellman
DSR	Dynamic Source Routing
E-HSAM	Enhanced Highly Secure Approach Method
eMANET	Emergency MANETs
GloMoSim	Global Mobile Information System Simulator
GPS	Global Positioning System
HSAM	Highly Secure Approach Method
MANETs	Mobile Ad Hoc Networks
MTT	Maximum Traversal Time
NetTT	Net Traversal Time

NodeTT	Node Traversal Time
RREP	Route Reply
RREQ	Route Request
RTT	Round Trip Time
TCP	Transmission Control Protocol
TDES	Triple Data Encryption Standard
TREP	Time of Reply
TREQ	Time of Request
TSMI	Timed and Secure Monitoring Implementation
UDP	User Datagram Protocol
WPT	Wormhole Prevention Timer

Chapter 1

Introduction

Mobile devices such as laptops and cellular phones are capable of sending data to each other on demand. This type of data transfer creates a temporary mobile ad hoc network (MANET) [1]. Unfortunately, MANETs do not have a centralized infrastructure providing security and are highly subjected to malicious attacks [2]. One example of an attack on MANETs is a wormhole attack. The consensus among the research community involved is that wormhole attacks aim to attack a fragile MANET by using two or more malicious nodes to fool a source node which is trying to send data. This is done by using a route which presents itself as the shortest route to the destination node [3]. A common wormhole attack involves a node which is used to record data and another node which is used to forward data back into the network [4]. In turn, there are other attacks which include the modification of data packets and thus the disruption of the integrity of the data as it travels in the network. Additionally, the dropping of data packets is also a possibility. Thus, the types of attacks are limitless.

1.1 Context of Our Study

This thesis investigates wormhole attacks in MANETs. The first scheme that is of interest to us is the HSAM scheme [5]. In this method, the data packets are split into 48 byte chunks

before being sent to the destination node. Moreover, the data frame includes a hash value for each chunk as well as a send time (time of delivery). The hash value helps to maintain the data integrity since a slight change in the data frame will result in a different hash value. Two counters are used to keep track of: (1) the number of packets sent to the destination node; and (2) the number of packets unsuccessfully delivered. After all the packet chunks have been sent, the ratio between (1) and (2) is used to determine whether the route is acceptable or compromised. The other protocol of interest is the AODV-WADR-AES protocol proposed in [6]. This scheme was developed to overcome wormhole attacks in emergency MANETs multimedia communications. In this method, the Advanced Encryption Standard (AES) and the Diffie-Hellman (DH) Key Sharing methods are used to prevent attackers from retrieving information from control packets while node-to-node timing analysis is performed. The goal of the method is to maintain security while increasing the network efficiency in terms of data packet delivery and end-to-end delay. However, this method does not take into account the security of data packets during their transfer. The aim of our proposed method is to address the shortcomings of the above mentioned schemes in order to design a scheme to protect against collaborative wormhole attacks in MANETs.

1.2 Motivation

Current threats against MANETs are becoming more and more sophisticated so that prevention solutions based on single attacks may no longer be sufficient. In MANETs, the identification of malicious activity is difficult when one node misbehaves during route formation. Further, if multiple malicious nodes collude together to perform malicious acts, their activity becomes even harder to detect. If multiple nodes act maliciously, simultaneously or alternatively to launch wormhole attacks, the schemes used to deal with them become less efficient and less effective at warding off these attacks. New solutions against these types of attacks are always desirable. This thesis attempts to propose a scheme which would be able

to efficiently and effectively address such attacks.

1.3 Research Problem

This thesis attempts to find a way to secure MANETs against collaborative wormhole attacks. There are many implementations which help deter single node attacks. However, not many existing solutions deal with collaborative attacks. Therefore, we would like to create a method to address the issue of collaborative attacks in MANETs. In addition to improving security, our aim is to increase the efficiency of the routing within MANETs. Our proposed method, TSMI, combines the strengths of HSAM [5] and AODV-WADR-AES [6] to achieve such goal.

The first implementation, HSAM [5], is a system which attempts to secure routes by splitting packets into chunks before sending data packets. Moreover, it creates a ratio which evaluates how secure the chosen route is. However, HSAM could still be subjected to collaborative attacks since each packet chunk contains some important information that could be copied by malicious nodes. In HSAM, the route is evaluated after the data chunks have been sent. If the route being used contains malicious nodes which are copying packet chunks, then it could be possible for the attackers to piece the packet chunks together and in turn, steal the original data packet. On the other hand, the scheme attempts to prevent tampering attacks by including hash markers with each chunk.

The idea of AODV-WADR-AES [6] is to monitor node to node traversal times (NodeTT), net traversal times (NetTT) and maximum traversal times (MTT) between source and destination nodes. Wormhole attacks typically have malicious nodes which are distant from each other and therefore, the nodes need to have a greater transmission range in order to tunnel packets to each other [2]. Thus, the NodeTT between the two malicious nodes will typically be higher than between normal nodes since the range is also greater than normal. In addition to monitoring the timing, AODV-WADR-AES also performs AES encryption on a secret key,

generated by the Diffie Hellman algorithm, which the source and destination nodes exchange. The results shown by [6] are impressive for a cryptographic algorithm. However, the method does not secure the sending of data packets after the route has been verified. Therefore, message integrity may be compromised when using AODV-WADR-AES. Moreover, AODV-WADR-AES is limited to only routes with 3-hops. Other routes with distances less than or greater than 3 are not considered and regular AODV protocols are used.

Our scheme (TSMI) aims to enhance both HSAM and AODV-WADR-AES and combine them together to create a more secure and efficient algorithm. We have designed E-HSAM (an enhanced HSAM) which addresses the aforementioned HSAM drawbacks. We have increased the security of HSAM by replacing the data packet chunks with mock packet chunks. We have also simplified the route blacklisting method which helped increase the efficiency of our algorithm. On the other hand, we have also enhanced AODV-WADR-AES by implementing a hash marker for data packets as they are sent through the route. By combining E-HSAM and AODV-WADR-AES, the advantages of each method target the weaknesses of the original methodologies. E-HSAM will secure all routes which are less than or greater than 3-hops between source and destination while AODV-WADR-AES will secure all 3-hop routes. Furthermore, both methods will share a common blacklist allowing a more efficient and secured routing in MANETs.

1.4 Thesis Contributions

The key contributions of this thesis can be summarized in the following points:

- We enhance a method called A Highly Secure Approach Method (HSAM) by introducing the Enhanced Highly Secure Approach Method (E-HSAM) [7] which improves the security and efficiency of HSAM while increasing the number of data packets through legitimate routes.

- As a complement to the AODV-Wormhole Attack Detection Reaction-AES (AODV-WADR-AES) [6] scheme which uses the Advanced Encryption Standard (AES), we implement the AODV-Wormhole Attack Detection Reaction-TDES (AODV-WADR-TDES) by using another cryptographic protocol called the Triple Data Encryption Standard (TDES).

- We propose a new scheme called A Timed and Secure Monitoring Implementation (TSMI). To do this, we combine the features of E-HSAM and AODV-WADR-AES. TSMI maintains the level of performance of both schemes while improving the number of packets received as well as achieving a higher delivery ratio.

1.5 Thesis Organization

- **Chapter 2** describes some background information on MANETs, AODV routing protocol, message delivery methods, cyptographic protocols and cipher techniques. Recent literature on wormhole attacks is also described.

- **Chapter 3** describes our methodologies for preventing collaborative wormhole attacks in MANETs.

- **Chapter 4** presents our simulation results.

- **Chapter 5** concludes our work and sketches further research directions.

Chapter 2

Background and Related Work

2.1 Background

2.1.1 MANETs and Wormhole Attacks

A MANET is a network structure without a centralized infrastructure, making it vulnerable to several types of attacks such as wormhole attacks. Wormhole attacks aim to attack a fragile MANET by using two or more malicious nodes to fool a source node, which is trying to send the data. Typically, the malicious node uses a fake route which presents itself as the shortest route to the destination node [3]. Usually, two nodes are involved: one is used to record the data and the other is used to forward the data back into the original network [4]. Contrary to malicious attacks, it was suggested [8] that if a wormhole attack is not used for malicious means (for instance, the case where it is created by security personnel to test a network), the performance of the network can be improved. An example of a wormhole link is depicted in **Figure. 2.1** [9]:

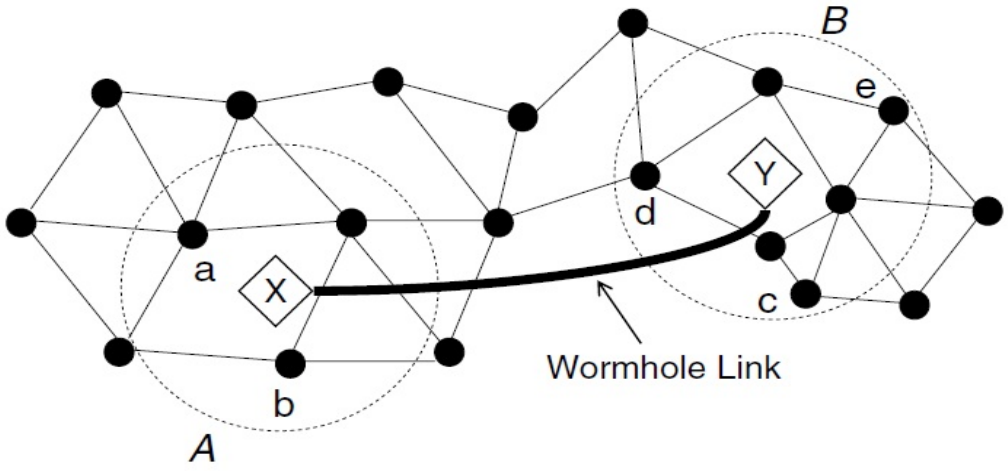

Figure 2.1: Wormhole Link

In **Figure. 2.1**, nodes X and Y are colluding nodes which form a wormhole link. These two nodes are part of the MANET and have the ability to communicate with each other as well as other normal, behaving nodes such as b or c. However, unlike behaving nodes, X and Y have a higher transmission power which gives them the ability to attract traffic to themselves, allowing malicious acts to take place.

2.1.2 AODV Protocol

AODV [10] is a routing protocol which has been designed to wait for requests before attempting to find the most optimal route for use by one node (source) to send messages to another node (destination). The most optimal route is determined by the distance or the number of hops between nodes. The AODV process works as follows:

1. A source node is required to signify the need to send packets to a particular destination node.

2. The source node broadcasts a route request packet (RREQ) to all neighbouring nodes.

3. The intermediate nodes continues broadcasting the RREQ to other neighbouring nodes until the destination node is found.

4. Once found, the destination node broadcasts a route reply packet (RREP) back to the source node using different routes found by the RREQ broadcast.

5. Once the source receives the RREP messages, it chooses the route with the lowest hop count to forward the data packets to the destination node.

2.1.3 Message Delivery Protocols

There are two types of message delivery protocols; namely, unicast and multicast. Unicast message delivery [11] is a type of message passing mechanism used in AODV. Unicast involves two nodes of interest, one node wanting to send a message and another node that will receive the message. Therefore, unicast is commonly known as 'one-to-one' message transmission. For example, after a source node receives a RREP from a destination node, it can only send the data packets to the destination node and no other nodes. Also, once the destination node receives a RREQ, it constructs a RREP and unicast the packet to the source node by retracing the hops of the RREQ.

Multicast message delivery [12] is another type of message delivery scheme. Unlike unicast, multicast can involve two or more nodes of interest. Therefore, multicast is commonly known as 'one-to-many' message transmission. For example, after a source node receives a RREP from different destination nodes of interest, it sends the data simultaneously to all the targeted destination nodes. In this thesis, only unicast transmission is used.

Since MANETs are open wireless networks which are vulnerable to attacks, we will need methods to protect messages and data packets which are passed through the network. The Diffie-Hellman (DH) algorithm [13] is ideal for use in AODV since it allows two nodes in the same network that are not aware of any predefined or known parameters of each other to securely exchange secret keys in an insecure network environment. We will be using the Advanced Encryption Standard [14] in CBC mode as the security protocol used to provide additional security of these secret keys.

2.2 Literature Review on Wormhole Attacks

Several works address the problem of detecting, preventing or mitigating wormhole attacks in MANETs. The following review represents a few of these solutions.

In [15], Khalil et al. proposed a protocol (so-called LiteWorp) to discover and prevent wormhole attacks in a static network. LiteWorp works by instructing each node involved to obtain 2-hop routing information from their immediate neighbours. The proposed technique is an extension of their original protocol in which each node only keeps 1-hop routing information. According to the authors, analyzing the routing information on routes which are 2-hops away will aid wormhole detection. In addition, nodes will also monitor their own neighbour nodes and potentially become guard nodes for a pair of nodes which are maximum 2-hops apart. Guard nodes are designed to monitor neighbouring nodes activity to aid wormhole detection. However, the LiteWorp idea was only designed for static networks and is impractical for a mobile network such as MANETs.

In [4], Hu et. al packet leashing introduces a way to prevent wormhole attacks. Their idea of using packet leashes consists of information in a commuting packet that detects and prevents abnormal transmission. In doing so, some timings are inserted in the packet when it is forwarded by the sending node to a receiving node (not necessarily from source to destination). Next, the receiver compares its own timings with the sender's timings resulting in a calculated latency between the two nodes. The authors pointed out a limitation to this technique since it is possible that two malicious nodes collude together to break their scheme while still being within an acceptable distance. They state that a network which includes node positioning analysis will overcome such vulnerability. Unfortunately, this solution suggests that only a network with such topology will be able to prevent this detrimental attack.

In [16], a scheme called TrueLink was proposed to prevent wormhole attacks. TrueLink also uses a timing based solution to the wormhole problem. Two nodes which are seemingly neighbours exchange nonces or values which are used only once. This method is monitored

under strict timing restrictions (similar to those used in [4]) so that it is impossible for colluding malicious nodes to replicate it since the packets routed through the wormhole will be more time consuming than those forwarded normally. After the exchange of nonces, the two apparent neighbours send messages to each other authenticating that the nonces were from the respective nodes. If any intrusion is discovered, there will be a mismatch in messages and the system will detect the malicious activity. This scheme suffers various limitations. First, unexpected delays caused by legitimate events such as signal interference may increase the timings past accepted ranges, thereby causing a drop of genuine packets. Second, it is assumed that an authenticating mechanism (which may not exist) is already in place. Third, if the security mechanism is not available, this method will be benign and ineffective to detect malicious attacks on the packets being forwarded within the network.

In [5], Mamatha et al. proposed an AODV-based scheme for preventing wormhole attacks in MANETs in which the hash identifier of the original packet is inserted in the data as it gets forwarded from node to node. The procedure of their scheme is outlined as follows.

1. The system gets the hash (treated as a marker) and includes it in the data frame of the packet.

2. The packet is then divided into sub-packets with a predetermined size and sent to the destination node.

3. The intermediate nodes read the data frame to get the destination of the packets and then continue forwarding to the next neighbouring node based on the AODV protocol.

4. Once the sub-packets are received by the destination, they are reconstructed to form the original packet and the hash value is calculated.

5. The receiver will continue to extract the hash value stored in the data frame of the packet and then checks if these values match. If that is the case, the receiver will reply with an acknowledgement (ACK) frame containing an ACK field to confirm safe

receipt of the packet. However, if there is no match, the ACK frame will contain a Confidentiality Lost field which will be forwarded to the sender informing it that the hash value was changed.

If the acknowledgement frame is not received by the sender within the allotted time, then the packets are assumed to be lost. Few limitations of this method are as follows. This method does not address other major effects of wormhole attacks. For instance, falsified nodes cannot be prevented from recording the packet chunks. Indeed, packet chunks can be copied or recorded without modifying the packet itself, thus the hash value will not change. Additionally, if packet chunks are recorded, they can possibly be reconstructed by malicious nodes. Thus, it is possible for a wormhole attack to record packets while remaining undetected.

In [17], Singh and Vaisla introduced an approach to detect wormhole attacks, where time is considered as a key parameter. In their scheme, during the RREQ broadcast phase of route discovery, each node will save a TREQ (time of request) which will record the time it takes for the current node to forward a RREQ to its neighbour node. Once the RREQ reaches the destination, a RREP is sent by the destination node to the sender and a TREP (time of reply) is recorded at each node as the algorithm retraces its steps back to the sender. Finally, a RTT (round trip time) of each successive intermediate node is calculated as the difference between the TREP and the TREQ (RTT = TREP - TREQ). The RTT is calculated at each node to check if the value is higher or lower than other RTT values calculated along the route. If no attack is detected, the value of the RTT will be similar for all nodes. However, if the value of the RTT at a certain node is higher than that of other successive nodes, then a wormhole attack may have occured at that specific node. The method assumes that the RTT between two colluding attacking nodes will be significantly higher than two good nodes. However, calculating only the RTT may not be sufficient to detect wormhole attacks.

In [9], a method is proposed that does not use timings or delays as a factor in detect-

ing wormhole attacks in wireless networks. Each node has a disk which its surrounding neighbours may populate. The method exploits the long range wormholes by using unit disk graphs (UDGs) as an unit radius disk in the network plane that models the range in which a node can communicate using an omni-directional antenna. Since long-range links will be more than a unit radius, this will indicate the presence of a wormhole link in the network. Although their proposed scheme was shown to effectively detect wormholes in wireless networks, the case when nodes are mobile was not treated.

In [18], Choi et al. proposed a method which is based on the DSR routing protocol to detect wormholes in MANETs. Their method consists of a neighbour node monitoring technique and a wormhole route detection method controlled by means of a timer (so-called wormhole prevention timer). This scheme does not rely on any specific hardware for node location or time synchronization. However, the assumptions that nodes will continue to monitor active transmission at the link layer even though some of these nodes may not be intended receivers is not realistic.

In [19], Hayajneh et al. proposed a technique called DeWorm which utilizes discrepancies in routing between neighbours of nodes that are along a route of a selected path between the source and destination nodes. DeWorm takes advantage of the fact that a wormhole link attempts to attract a large amount of traffic to itself. Moreover, the routes through the wormhole link are shorter than that of legitimate nodes within the network. Each node along the route, after being selected during the route discovery phase, will initiate the DeWorm algorithm which relies on the acquisition of different routes to a target node. This route discovery phase involves nodes which are 1-hop neighbours of the current node performing the scheme. Additional information has been added to each packet in order to carry necessary information used by the method. Therefore, this method suffers from a large overhead. Due to this large overhead, the computation used to analyze the necessary information will increase, thereby increasing the amount of energy consumed. Since the method is not energy efficient, this protocol is not quite suitable for mobile networks for

which the target is energy conservation.

In an attempt to improve DeWorm, Dhurandher et al. [20] proposed a scheme (so-called E^2SIW) which is based on the AODV protocol. It is designed to be energy efficient and can mitigate wormhole attacks in MANETs. Their method uses the location coordinates of nodes within the MANET in order to detect the presence of wormholes. The authors discussed that wormhole links will be low in hop counts. The hop count needs to be low in order to attract a large amount of traffic to the wormhole nodes. By using GPS to examine the connectivity information for all neighbouring nodes of a given node, the method is able to detect exactly how far the nodes are from each other. Moreover, as the RREQ is being broadcasted, each hop is analyzed by their respective nodes. Information about nodes which are two hops away are tested and temporary routes are analyzed as well. Thus, this technique continually monitors the actual position of nodes as well as the temporary routes during the route discovery phase. By detecting anomalies within the network throughout and each hop of the discovery process, the method continuously builds a route which avoids the wormhole link. However, the authors have not disclosed the overhead for such a scheme since a lot of information is analyzed throughout each step of the route discovery process.

Chapter 3

Methodologies

In this chapter, we describe the methods that we proposed to address wormhole attacks in MANETs as follows: (1) we analyze two different single wormhole scenarios and its effects on MANETs by injecting two colluding malicious nodes which drop data packets; (2) we introduce our E-HSAM method to enhance the HSAM scheme. The enhancements are twofold (a) performing for the first time a simulation study of the HSAM protocol, (b) introduceing a more efficient approach to securing the routes in the route selection phase of the HSAM protocol; (3) we design AODV-WADR-TDES as a complement to AODV-WADR-AES in order to test the performance difference when using two different cryptographic algorithms with the same AODV-WADR scheme. This study is valuable because there exists modern day technology, such as many smart cards, which do not necessarily support AES [21]; (4) we combine the techniques and strengths of both E-HSAM and AODV-WADR-AES to design our novel scheme (so-called TSMI). TSMI is able to ward off collaborative wormhole attacks effectively while maintaining the performance of both E-HSAM and AODV-WADR-AES.

3.1 Prevention Against Single Wormhole Attacks

3.1.1 Introduction

To begin our study of MANETs, we need to examine the effects of single wormhole attacks before investigating collaborative attacks. We developed two simple scenarios where a MANET is affected by a single wormhole attack. Both scenarios involve a wormhole which targets the integrity of data packets.

3.1.2 Single Wormhole: Method 1

In this scenario, we simulate two different single wormhole attacks. In order to do this, we must define the two colluding or malicious nodes. In this simulation, we use nodes 6 and 10 as the two colluding nodes. The idea is to get these two nodes to have direct communication with each other. For example, if node 10 gets an RREQ packet, the next hop should be to the other malicious node, namely node 6 and vice versa. It is important to note that the radio power of the two colluding nodes is higher than that of normal nodes. The reason for increasing the power of the two nodes is part of what is called a long range wireless attack. Since the two nodes have a higher range, the number of neighbours to the nodes will increase, thus reducing the number of hops required to get to the destination node. Malicious node 10 drops data packets as they are tunneled through the wormhole. **Figure. 3.1** displays an example of how method 1 works:

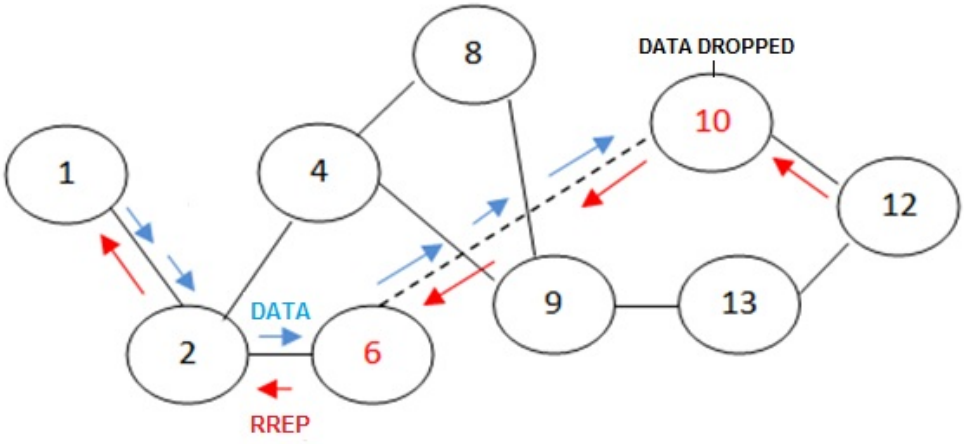

Figure 3.1: Single Wormhole Attack Method 1

3.1.3 Single Wormhole: Method 2

In this simulation, we also use nodes 6 and 10 as the two colluding nodes. This scenario is slightly different than that in method 1. Although malicious nodes 6 and 10 are colluding nodes, packets are only forwarded from node 6 to node 10 but not the other way around. Node 10 will not forward any packets to node 6 as in the first method. Another important constraint of this scenario is that one of the colluding nodes must be a direct neighbour of the source node (here, malicious node 6) and the other node must be a direct neighbour of the destination node (here, malicious node 10). As with method 1, the radio power of the two malicious nodes is higher than that of normal nodes. Finally, the attack scenario of this method is the same as in method 1; that is, data packets are dropped by the mailicous node 10. Method 2 is depicted in **Figure. 3.2**.

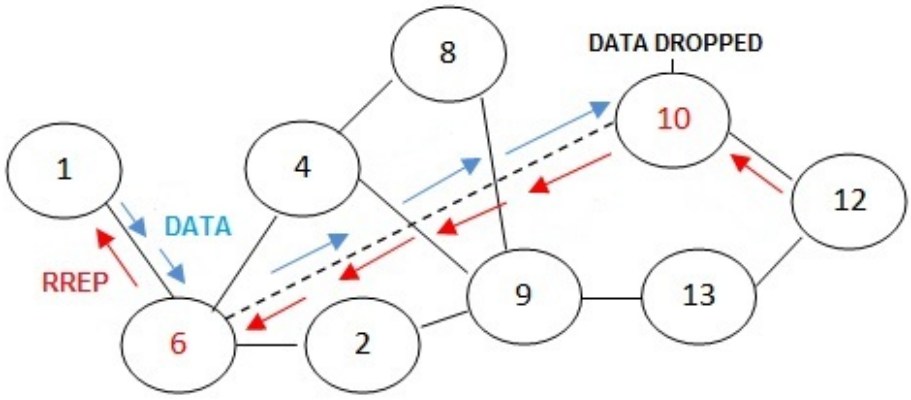

Figure 3.2: Single Wormhole Attack Method 2

3.2 Prevention Against Collaborative Wormhole Attacks

3.2.1 A Secured Routing Protocol for Preventing Packet Dropping and Message Tampering Attacks

In this section, we introduce an enhancement to an existing secured routing protocol [5] against packet dropping and message tampering attacks in MANETs called the Highly Secured Approach against Attacks in MANETs (HSAM). Our goal is to improve the data integrity part of HSAM. The focus of the method is to create a dynamic methodology which tests a selected route as data packets are being sent when using the AODV protocol within MANETs. Before a data packet is about to be sent to the desired destination node, the algorithm splits the packet into packet chunks and calculates a hash marker for each chunk. Once in chunks, each sub packet is sent to the destination. If the hash value is incorrect, the route is blacklisted. The number of data chunks sent as well as unsuccessful transmissions are recorded and a tolerance ratio is calculated. After all the chunks are sent, the original data packet is reconstructed at the destination node. Although HSAM deals with the integrity of the data packet, there are still attacks which makes this method vulnerable. For instance,

the copying of packets is not addressed, i.e. malicious nodes could potentially record all the data packet chunks and assemble them together to get the original packet.

3.2.1.1 Highly Secured Approach Against Attacks in MANETs

As mentioned, the main focus of the Highly Secured Approach Against Attacks in MANETs (HSAM) method is to create more security for data packets as they are being forwarded within a MANET. Route discovery is obtained using the regular AODV methods (i.e. sending out RREQs and receiving RREPs). Once a route is ready, data packets are first divided into 48 byte data frames. Information about the source node and destination node as well as a hash marker and time of delivery is inserted into the data frame. The hash value will help maintain the data integrity since a slight change in the data frame will result in a different hash value. After the data frame is sent, a counter (cpkt) will be incremented to keep track of the number of chunks sent. The authors referred to this technique as one way hash chain. They also argued that it will be impossible for an attacker to break it. Next, intermediate nodes will forward the data frame until it reaches the destination. Once the destination node receives the data frame, the hash will once again be calculated and matched against the hash code obtained from the data frame. One of the following two different acknowledgement frames or ACK frame will be created by the destination node and sent back to the source node:

- The hash code matches the code within data frame. An ACK frame containing an ACK field, which indicates that the hash code is verified and secure, is sent back to the source node.
- The hash code does not match the code within the data frame. In this case, an ACK frame containing a Confidentiality Lost field, which indicates that the hash code has been compromised, is sent back to the source node.

Once the ACK frame is received by the source node:

- If the integrity is verified (i.e. ACK field is received), the receipt time is calculated and recorded. If the difference between the send and receipt time is equal or less than 20ms, then the route is deemed to be secure and the next 48 byte data chunk will be sent. However, if the time is greater than 20ms, the packet is assumed to have been lost and a counter (cmiss) will be incremented in order to keep track of the number of conceivably missed packets.

- If the hash code does not match with the one within the data frame (i.e. Confidentiality Lost field received), the route is discarded, blacklisted and the next route is chosen using regular AODV methods

Assuming that the packet integrity is validated and all the packet chunks have been sent, a ratio called the Limit of Tolerance will be calculated. The Limit of Tolerance is based on the Principle Flow of Conservation [22] in order to measure the forwarding misbehaviour of nodes. The ratio, cmiss/cpkt, that is, the number of missed packets over the number of packet chunks sent, will be calculated and a limit of 0.2 or 20% will be set for its value. If this ratio is equal or less than 0.2, the route will be deemed safe. Otherwise, the route will be considered as insecure and the next available route will be used to send the next set of packet chunks.

The HSAM algorithm pseudocode (here referred to as **Algorithm 1**) is shown below. In **Algorithm 1**, the following notations are used:

- RREP: Route Reply

- RTT: Round Trip Time

- RREQ: Route Request

- ACK: Acknowledgement

- NetTT: Net Traversal Time

Algorithm 1 HSAM

Require: AODV Protocol

1: Route (R) is obtained through the regular AODV protocol
2: RREQ sent out by source (S) and S records the current time (t)
3: **if** S receives RREP from destination (D) within NetTT **then**
4: S records the receive time duration (t) and hop count found in RREP
5: **else**
6: S selects the next best route available from RREPs
7: **end if**
8: S divides the data packet into sub-packets of 48 byte chunks each
9: **for** each packet chunk to send **do**
10: Construct the data frame with source address, destination address and hash code
11: Increment the counter cpkt
12: Send packet to nearest node and record the send time in RTT_{start}
13: Packet reaches D and the hash is computed and compared to the hash in the frame
14: **if** there is a match (in hash values) **then**
15: D prepares ACK frame with ACK field and sends it to S
16: **else**
17: D prepares ACK frame with CONFIDENTIALITY LOST field and sends it to S
18: **end if**
19: **if** frame contains the ACK field **then**
20: S records the receive time in RTT_{end} and calculates the RTT as RTT_{start} - RTT_{end}
21: **if** RTT is equal or less than 20ms **then**
22: Route is valid and S prepares to send the next chunk
23: **else**
24: Packet is assumed to be lost and S increments the counter **cmiss**
25: **end if**
26: **else**
27: S blacklists and discards R, then selects next route R from line 1
28: **end if**
29: **end for**
30: S calculates the ratio cmiss/cpkt to obtain the Limit of Tolerance value
31: **if** ratio is less than or equal to 0.2 (20%) **then**
32: R is not discarded from the routing table and the route is deemed acceptable and secure
33: **else**
34: R is compromised, S blacklists R and selects next R from line 1
35: **end if**

All single operations in the HSAM algorithm are approximately O(1). The need to send all the sub-packets to the destination node costs approximately O(n) where n is the number of packets to be sent. Therefore, the complexity of this algorithm is approximately O(n) + O(1) or O(n).

3.2.1.2 Enhanced HSAM Scheme

We propose an enhancement to the HSAM protocol (so-called E-HSAM). The implementation of E-HSAM is similar to that of HSAM; however, the major difference is found in the sending of packets. HSAM splits the data packets into 48 byte chunks and sends them to the destination. However, it takes a reactive approach to securing the routes. Data chunks may still contain, in part, sensitive data which can still be stolen or tampered by intermediate malicious nodes before the route is discarded. In an attempt to eliminate any chance of unwanted data manipulation or data copy, we propose that mock packets be sent instead of the actual data chunks. The mock chunks contain filler content which is not part of the original data packet. Therefore, if an attack occurred, the actual data packet will not be compromised. In our simulation, the number of mock packets sent is obtained by dividing the payload size of the actual packet by a split value of 48. The reasoning behind this comes from our treatment of the size of data packets. In order to obtain a reliable Limit of Tolerance, there is a need to obtain a sufficient number of cpkt and cmiss. For smaller data packet sizes, the packet should be split accordingly in order to gather enough cpkt and cmiss to provide a reliable ratio (rather than just a few cpkt and cmiss). The next difference is how routes will be avoided by the method. Instead of using a self-developed method to avoid the route containing malicious nodes, E-HSAM method utilizes a mechanism similar to that used by AODV for the sending of a RERR packet back to the sender. This mechanism will discard the suspicious route and automatically increment the routing table sequence number, then choose the next route. This slight modification to the RERR mechanism effectively avoids the route in question when data the integrity is compromised and the next available route is to be used. **Algorithm 2** describes the E-HSAM algorithm pseudocode:

Algorithm 2 E-HSAM

Require: AODV Protocol

1: Route (R) is obtained through the regular AODV protocol
2: RREQ sent out by source (S) and S records the current time (t)
3: **if** S receives RREP from destination (D) within NetTT **then**
4: S records the receive time duration (t) and hop count found in RREP
5: **else**
6: S selects the next best route available from RREPs
7: **end if**
8: ***S creates mock packets of 48 bytes each***
9: S divides the data packet into sub-packets of 48 byte chunks each
10: **for** each packet chunk to send **do**
11: Construct the data frame with source address, destination address and hash code
12: Increment the counter cpkt
13: Send packet to nearest node and record the send time in RTT_{start}
14: Packet reaches D and the hash is computed and compared to the hash in the frame
15: **if** there is a match (in hash values) **then**
16: D prepares ACK frame with ACK field and sends it to S
17: **else**
18: D prepares ACK frame with CONFIDENTIALITY LOST field and sends it to S
19: **end if**
20: **if** frame contains the ACK field **then**
21: S records the receive time in RTT_{end} and calculates the RTT as RTT_{start} - RTT_{end}
22: **if** RTT is equal or less than 20ms **then**
23: Route is valid and S prepares to send the next chunk
24: **else**
25: Packet is assumed to be lost and S increments the counter **cmiss**
26: **end if**
27: **else**
28: S blacklists and discards R, ***broken links are incremented, S selects the next R from the routing table and restarts the mock packets process (i.e. repeat line 8)***
29: **end if**
30: **end for**
31: S calculates the ratio cmiss/cpkt to obtain the Limit of Tolerance value

32: **if** ratio is less than or equal to 0.2 (20%) **then**
33: R is not discarded from the routing table and is deemed acceptable and secure
34: ***The real data packet and hash code are sent to the destination node using route R***
35: **else**
36: S blacklists and discards R, ***broken links are incremented, S selects the next R from the routing table and restarts the mock packets process (i.e. repeat line 8)***
37: **end if**

E-HSAM addresses collaborative attacks which exists in the original HSAM. By sending mock packet chunks, if an attacker tries to copy the packets, the original data is not revealed since the mock chunks do not contain any information which could be useful to the involved malicious nodes. Additionally, the packets are protected from tampering attacks since hash markers are utilized to identify changes to the mock chunks as well as the actual data.

All single operations in the E-HSAM algorithm are approximately O(1). The need to send all the sub-packets to the destination node costs approximately O(n) where n is the number of packets to be sent. Therefore, the complexity of this algorithm is approximately O(n) + O(1) or O(n).

3.2.1.3 E-HSAM-AES Scheme

In E-HSAM, a hash marker is used for each mock packet chunk in order to detect tampering attacks resulting from wormhole links. Although adding hash values to a packet provides a level of security to packet tampering attacks, it is possible to create a stronger safety measure for the packet. This can be done by embedding the AES algorithm within E-HSAM (so-called E-HSAM-AES). Although a hash marker is embedded into each mock packet, the acknowledgement frame that is sent back from the destination node does not contain a hash marker. Therefore, it is possible to tamper with the marker with the acknowledgement (ACK) frame without triggering the security mechanism in E-HSAM. For instance, let us assume that the colluding nodes are able to successfully tamper the mock packets as they traverse towards the destination node. The destination node will be alerted of this fact by

the different hash marker and will prepare an ACK frame with a Confidentiality Lost field. As the ACK frames travels back to the sender with the same route, the colluding nodes can change the Confidentiality Lost field to the accepted ACK field. Since there is no hash marker accompanying the ACK frame, the sender will not be able to realize that the message has been tampered with and will assume that the initial transfer is successful. Assuming the wormhole route is accepted and the actual data packets are sent, the malicious nodes will be free to tamper with these data packets.

To further increase the security of our scheme, we combine the hash marker with a cryptography method in order to effectively mask the packets. Here, cryptography is not necessarily applied in the phase where the selected route is evaluated for its suitability to forward the data (line 22 in **Algorithm 2**) and when the limit of tolerance is calculated (line 31 in **Algorithm 2**) since our technique uses mock packets. By choosing AES as the cryptographic scheme, we aim to further protect data packets as they traverse through the MANET in route to the destination node. **Algorithm 3** describes the pseudocode of the E-HSAM-AES algorithm.

Algorithm 3 E-HSAM-AES

Require: AODV Protocol

▷ **Lines 1 to 30 remain the same as E-HSAM**

31: S calculates the ratio cmiss/cpkt to obtain the Limit of Tolerance value
32: **if** ratio is less than or equal to 0.2 (20%) **then**
33: R is not discarded from the routing table and is deemed acceptable and secure
34: ***The real data packet and hash code (encrypted with AES) are sent to the destination node using route R***
35: **else**
36: S blacklists and discards R, broken links are incremented, S selects the next R from the routing table and restarts mock packets process (i.e. repeat line 8)
37: **end if**

All single operations of the E-HSAM-AES algorithm are approximately O(1). The need to send all the sub-packets to the destination node costs approximately O(n) where n is the number of packets. Therefore, the complexity of this algorithm is approximately O(n) +

O(1) or O(n).

3.2.2 Securing AODV Against Wormhole Attacks in Emergency MANET Multimedia Communications

In this section, we examine the method of Securing AODV against Wormhole Attacks in Emergency MANET Multimedia Communications (so-called AODV-WADR-AES) [6]. AODV-WADR-AES is a scheme which uses the Advanced Encryption Standard (AES) as the cryptographic algorithm as well as nodal timing analysis to aid in preventing wormhole attacks.

3.2.2.1 AODV-WADR-AES Scheme

In [6], Panaousis et al. introduced AODV-WADR-AES to overcome and mitigate wormhole attacks in emergency MANETs (eMANETs) multimedia communications. In their scheme, the AES and Diffie-Hellman (DH) key exchange methods are used to deter attackers from retrieving information from control packets while node to node timing analysis is performed. The reason for using the DH key exchange is to help nodes authenticate each other in an insecure network [13]. AODV-WADR-AES begins by obtaining routes using the normal AODV protocol. The method then proceeds by extracting the hop count from the RREP during the route discovery. If the route obtained is not 3-hops, then normal AODV routing protocol is initiated. Assuming that a 3-hop route has been selected, the source node attempts to authenticate itself along with the destination node using an AES encrypted DH key along with timing parameters on the messages used during identity verification. If authenticated, the source begins transferring the data packets to the destination. However, there are some drawbacks associated with this method: (1) the data packets are not protected during their transmission using AODV-WADR-AES. For example, if a node is hijacked by a malicious user, it is possible for the user to tamper with data packets; (2) AODV-WADR-AES is restricted to analyzing 3-hop routes only. Even though the authors

claimed that this requirement is sufficient, it is possible that wormhole links exist in routes that have less or more than 3-hops. For instance, if a MANET is initiated and a wormhole link exists on a route which has more than 3-hops, the algorithm will not be used to analyze this route since the 3-hop requirement will be violated. Therefore, it is possible for malicious nodes to perform some of their deeds before being detected by the algorithm.

The following are some definitions used within the the algorithm.

- **Net Traversal Time (NetTT) in milliseconds (ms)**: this defines the maximum time allotted for receiving a RREP from the destination node after the source has broadcasted a RREQ
- **Node Traversal Time (NodeTT) in milliseconds (ms)**: this defines the maximum time allotted for packets to travel between each hop
- **Actual Traversal Time (ATT) in milliseconds (ms)**: this defines the actual calculated time for receiving a RREP from the destination node after the source has broadcasted a RREQ
- **Actual Traversal Time WADR (ATT_WADR) in milliseconds (ms)**: this defines the actual calculated time for receiving a modified AODV_WADR packet from the destination after the source has sent an original AODV_WADR packet
- **Maximum Traversal Time (MTT) in milliseconds (ms)**: this is defined by **6 x NodeTT** since only destinations that are 3 hops away are considered by this method

The main focus of the AODV-WADR-AES approach is to maintain the security of the data packets as they are being forwarded within an AODV-based MANET. One main requirement of this approach is that the destination node should be 3-hops away from the source node. The authors [6] argued that this restriction is sufficient to prevent wormhole attacks for two reasons:

1. If a source node (S) detects and prevents a wormhole, any routes passing through S will be diverted away from the wormhole.

2. It will be difficult for S, which is more than 3-hops away from the destination, to know which node in the route is malicious since S will only keep the information about the next hop node (which is one main features of AODV)

AODV-WADR-AES begins with small additions to the normal operations of the AODV protocol as follows. A source node that is in need of a route to send its packets to a destination node will broadcast a RREQ and will record the time which the request was made. When the source node receives a RREP, it will record the receipt time of the RREP packet. The two recorded times are used to calculate the actual traversal time (ATT) which will be used compared against the net traversal time (NetTT) to determine whether the algorithm will be launched or not. Indeed, if ATT is greater than NetTT, AODV-WADR-AES will be ignored and the normal AODV operations will commence. If ATT is less than NetTT and the hop count is not 3, AODV will operate normally. If ATT is less than NetTT and the hop count is 3, AODV-WADR-AES will be launched.

The next step is to check if the ATT is greater than the maximum traversal time (MTT). If ATT is less than MTT, the source node will consider that the route is safe and the normal AODV operations will continue. If ATT is greater than MTT, there is a greater chance that a wormhole may exist in the network (since the network will experience an increase in latency between nodes).

Once a wormhole is suspected, a shared secret key is created between the source and destination nodes. The DH method is used to determine this key. The DH algorithm is ideal for use in AODV since it allows two nodes which are not aware of any predefined or known parameters of each other to securely exchange secret keys in an insecure network environment [13].

The source node will first send a message to the destination node, stating that DH key exchange is needed and records the send time. Once a reply is received from the destination

node, the source node will record the receipt time and compare it to the send time. If ATT of the message is greater than NetTT, a wormhole attack is suspected and the source will delete the route from its routing table. Additionally, the source will update its blacklist (blacklist_wa) with the next hop node. If ATT of the message is less than NetTT, the DH key exchange will commence.

Once the DH key is created, the source node will then send an encrypted message (msg_wadr) to the destination node and record the time. AES is used for the encryption of msg_wadr. The destination node will decrypt msg_wadr, add its unique ID number associated with itself to it, re-encrypt msg_wadr and send it back to the source node. The source node will then record the time of receipt and calculate the actual traversal time of receiving a msg_wadr reply from the destination, i.e. ATT_WADR. If ATT_WADR is greater than MTT, a wormhole attack is suspected and the source node will delete the route from its routing table and will update its blacklist table with the next hop node of the route. If ATT_WADR is less than MTT, the source node will consider that the route is safe and AODV will be performed normally.

Algorithm 4 describes the pseudocode of the AODV-WADR-AES algorithm.

Algorithm 4 AODV-WADR-AES

Require: AODV Protocol

```
RREQ sent out by source (S) to destination (D) and S records the current time (t)
S checks the hop count when RREP is received and records the time (t') of receipt
if S receives an RREP within NetTT then
    if hop count is equal to 3 then
        S calculates ATT (t' - t)
        if ATT is higher than MTT then
            S sends message to D to begin DH key exchange and records time (t_DH)
            S receives reply from D and records the time (t'_DH)
            if ATT (t'_DH - t_DH) is less than NetTT then
                DH key exchange commences and secret key created
                S sends AES encrypted message (msg_wadr) containing the secret key to
                D and records the time (t_aes)
                D decrypts msg_wadr, adds its unique ID number to msg_wadr, re-encrypts
                it and sends it back to S
                if S receives msg_wadr then
                    S records receiving time (t'_aes) and calculates ATT_WADR (t'_aes - t_aes)
                    if ATT_WADR within MTT then
                        Route is considered safe and the normal AODV operations
                        commence
                    else
                        A wormhole is suspected, S deletes the route R from its routing
                        table and updates its blacklist (blacklist_wa) with the next hop node
                        of the suspect route
                    end if
                else
                    A wormhole is suspected, S deletes R from its routing table and updates
                    its blacklist (blacklist_wa) with the next hop node of the suspect route
                end if
            else
                A wormhole is suspected, S deletes R from its routing table and updates
                its blacklist (blacklist_wa) with the next hop node of the suspect route
            end if
        else
            Route is considered safe and normal AODV operations commence
        end if
    end if
    Normal AODV operations commence
else
    Normal AODV operations commence
end if
```

All single operations are approximately O(1). The need to forwarding messages and packets to the destination node costs approximately O(n) where n is the total number of control packets as well as AODV-WADR-AES messages. Therefore, the complexity of this algorithm can be measured to approximately O(n) + O(1) or O(n).

3.2.2.2 AODV-WADR-TDES Scheme

AES encryption is widely accepted as the next generation encryption technique [23]. However, technologies such as smart cards, which can be considered as eMANET nodes, are incompatible with AES implementations [21]. In light of this incompatibility, a different encryption technique called the Triple Data Encryption Standard (TDES) is recommended by [24, 25]. As a complement to the AODV-WADR-AES scheme, we introduce AODV-WADR-TDES. The methods found in AODV-WADR-TDES are similar to AODV-WADR-AES. However, the cryptographic algorithm of choice is TDES. By implementing longer variable key sizes to DES, TDES effectively solves the small key size problem by allowing 64, 128, 192-bit key sizes with an effective 56, 112, 168-bit key size respectively. With three different possible key combinations, the pseudo-randomness of the algorithm increases, thereby increasing its security potential.

The AODV-WADR-TDES protocol follows similar steps as the AODV-WADR-AES protocol. However, some differences occur in the manner the data packets are delivered, as well as the cryptographic scheme used. In the AODV-WADR-AES scheme, data packets are sent throughout the network after the constructed route has been deemed safe to be used. In the AODV-WADR-TDES, a hash algorithm is invoked that determines a hash value to be included with the data packets and to be used with TDES (in lieu of AES as cryptographic algorithm) to strengthen the integrity and confidentiality of the data packets.

3.2.3 A Novel Timed and Secured Monitoring Implementation Against Wormhole Attacks

With E-HSAM, we have successfully improved the HSAM scheme in order to address some potential security holes. With AODV-WADR-AES, we have overcome the limitation of dealing with routes where the source node is only 3-hops away from the destination node. However, AODV-WADR-AES does not protect the data packets during data transmission. In light of these shortcomings, we propose a new method called A Timed and Secure Monitoring Implementation against Wormhole Attacks in MANETs (so-called TSMI) that benefits from the features of both E-HSAM and AODV-WADR-AES to address some of their weaknesses. In TSMI, E-HSAM will be monitoring all routes which are not 3-hops between the source and destination nodes and AODV-WADR-AES will be handling all 3-hop routes. A hash value is also added to each data packet in AODV-WADR-AES in order to increase the integrity of the data transmission. Moreover, in TSMI, a blacklist is also introduced in order to quickly sort out malicious nodes that are caught by either technique (E-HSAM or AODV-WADR-AES). The TSMI algorithm is described as follows.

Algorithm 5 TSMI

Require: AODV Protocol

```
RREQ sent out by source (S) to destination (D) and S records the current time (t)
if S receives an RREP then
    S checks the hop count and records the time (t') the RREP was received
    if hop count is 3 then
        S calculates ATT (t' - t)
        if ATT is higher than MTT then
            AODV-WADR-AES algorithm will commence
        else
            Regular AODV operations will commence and the hash value will be included
            in the data packet
        end if
    else
        E-HSAM algorithm will commence
    end if
else
    E-HSAM algorithm will commence
end if
```

The complexity of E-HSAM is approximately O(n) and AODV-WADR-AES is approximately O(n). In TSMI, a target route will either be 3-hops or a different hop count. Moreover, the procedures of E-HSAM is independant of AODV-WADR-AES. Therefore, the complexity is based on the number of hops of the route. However, since both schemes have the same complexity, the complexity of TSMI is approximately O(n).

Chapter 4

Performance Evaluation

This chapter discusses all observations found for all simulations which have been evaluated throughout this thesis. All scenarios use the same tools and operating environments: the simulation tool is GloMoSim 2.03; the operating system is a 32-bit Linux based system; the compiler is Parsec with RedHat-7.2 configuration files.

All scenarios were run through different simulation trials. Additionally, we simulated each trial 3 times to ensure the accuracy of the acquired data. The main traffic generator used in all simulations is Constant Bit Rate (CBR). The message delivery protocol of our MANET is unicast delivery.

4.1 Single Wormhole Attacks

4.1.1 Assumptions and Scope of the Simulations

We assume the movement and direction of each node is randomized by the simulator and each node is restricted to move within the specified terrain of the MANET. Further, the change in speeds of our nodes will be restricted between 10m/s (meters/second) and 90m/s with 10m/s increments. Our single wormhole attack scenarios cover two different possible attacks. The idea is to get an introductory understanding of how malicious wormhole links

affect the MANET operations. It is also assumed that malicious nodes in our MANET will drop packets as they traverse through the wormhole.

4.1.2 Simulation Parameters

Table. 4.1 outlines the main parameters of our simulation for the studied single wormhole attack scenarios:

Table 4.1: Single Wormhole Attack Parameters

Routing Protocol	AODV
Traffic Type	CBR
MAC Protocol	IEEE 802.11
Mobility Model	Random Way Point
Number of Nodes	100
Default Wireless Power (txPower_dBm)	15 dBm
Malicious Wireless Power (txPower_dBm)	40 dBm
Max Movement Speed	10-90 m/s with 10 m/s increments
Simulation Terrain (m x m)	1000 x 1000
Simulation Time	1500 Minutes
Packet Size	512 Bytes

4.1.3 Performance Metrics

In order to measure efficiency and performance, we consider the following performance metrics:

- Number of packets received by the destination node
- Average end-to-end delay between the source and destination nodes: This is the average latency in milliseconds when sending a packet from a source node to destination node

4.1.4 Simulation Scenarios

As mentioned in Chapter 3, we will be evaluating two different single wormhole attack scenarios, Method 1 and Method 2. Method 1, depicted in **Figure. 3.1**, deals with a typical

bi-directional wormhole attack where the two colluding nodes are passing packets directly towards each other (i.e. if node 6 gets a packet, it will pass the packet directly over to node 10 and vise versa). The second method shown in **Figure. 3.2**, contains a unilateral wormhole; that is, only one mal node will direct packets to the other malicious node (i.e node 6 directs packets to node 10 but packets received by node 10 will not be directed to node 6. Additionally, each malicious node must be direct neighbours of the source and destination nodes (i.e. node 6 must be a neighbour of the source node and node 10 must be a neighbour of the destination node). In both scenarios, node 10 will drop data packets once the wormhole link route has been established. Both malicious nodes will have increased radio ranges (txPower_dBm of 40) which is 2.67 times greater than the regular default node radio strength (txPower_dBm of 15).

4.1.5 Results on Single Wormhole Attacks

4.1.5.1 Method 1

In this experiment, we measure the above mentioned performance metrics with respect to mobility of nodes. First, we observe the number of packets which are received by the destination node in **Figure 4.1**.

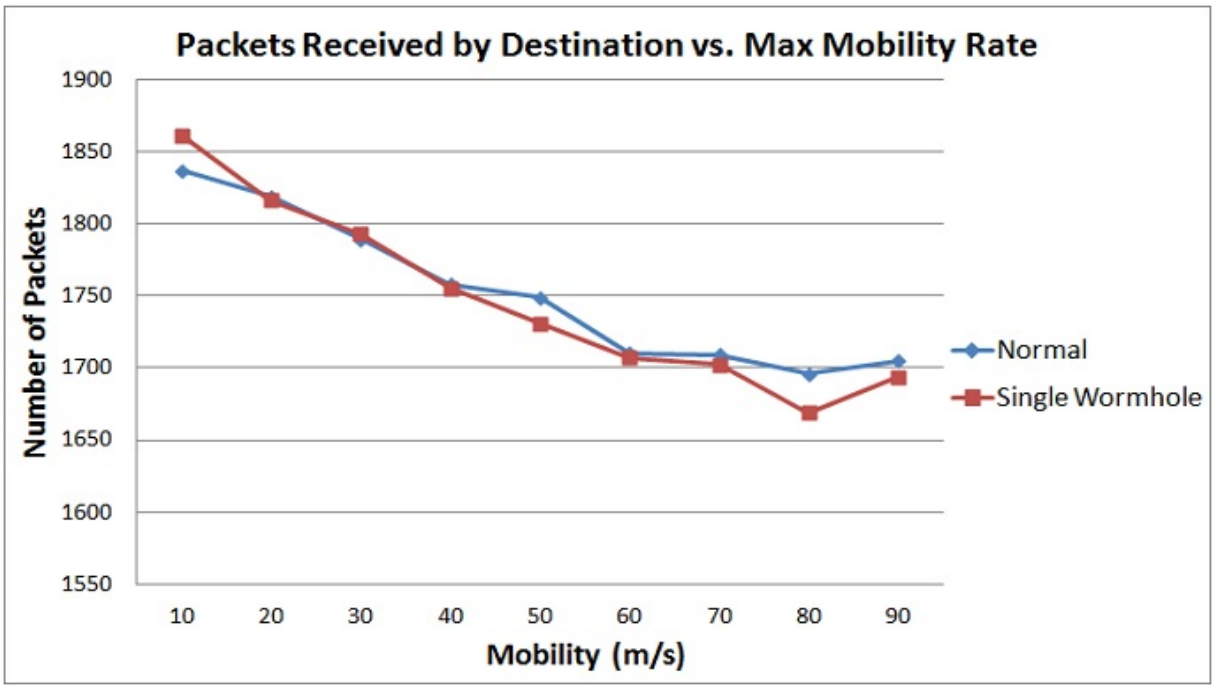

Figure 4.1: Method 1 - Packets Received by Destination vs. Max Mobility

As depicted in **Figure 4.1**, the number of packets reaching the destination node in most of the trials is lower when the network is experiencing a single wormhole.

Next, we examine the effect on the average end-to-end delay on a MANET (**Figure. 4.2**). Since the end-to-end delay measures the time it takes a packet to reach a destination node from the source node, it is an important metric to analyze.

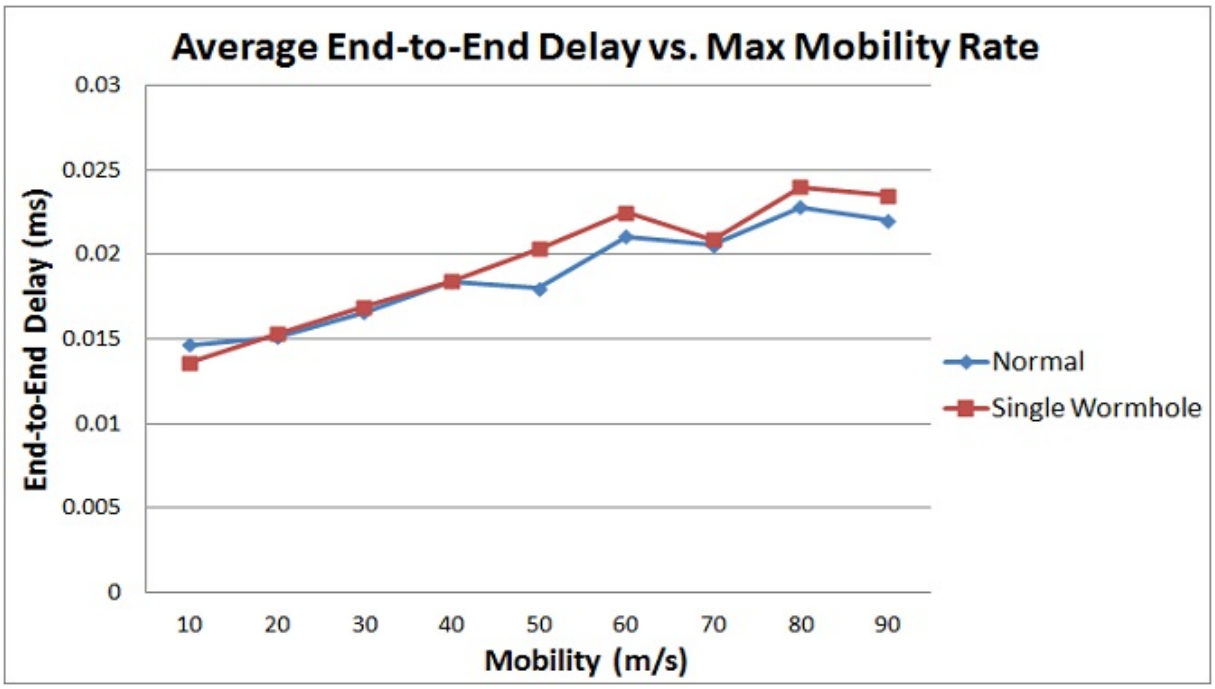

Figure 4.2: Method 1 - Average End-to-End Delay vs. Max Mobility

In **Figure. 4.2**, it can be observed that the average end-to-end delay is more prominent when the network is involved with a single wormhole attack. The delay increases as the mobility rate increases because of the increased mobility of nodes. As mobility increases, the number of broken links will also increase, thereby creating more delays when trying to establish a connection. Such result is also reflected in the observations for the number of bytes and packets sent to the destination. As mobility increases, due to more broken connections, the number of bytes and packets received will also decrease. The observed result is an increase of **3.6%** in average delay.

4.1.5.2 Method 2

In this slightly different scenario, we analyze the effect of the wormhole link under the same metrics used for Method 1. In Method 2, both malicious nodes must be direct neighbours or the source and destination nodes. Here, node 6 must be a direct neighbour of source node 1 and node 10 must also be a direct neighbour of destination node 12. We examine the

number of packets that the destination node has received from the source node. The results are depicted in **Figure. 4.3**.

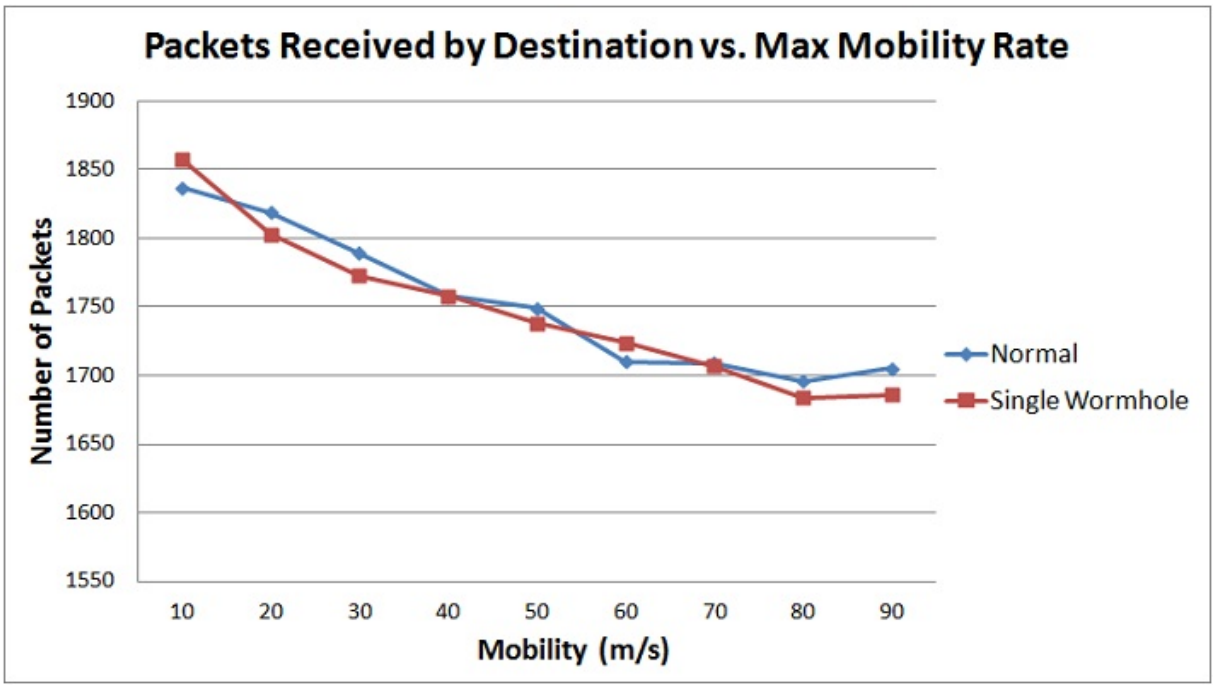

Figure 4.3: Method 2 - Packets Received vs. Max Mobility

The number of packets reaching the destination node (**Figure. 4.3**) for some trials is lower when the network is experiencing a single wormhole. However, there is also an increase of packets received with certain mobility speeds. The number of packets received by the destination node decreases as the mobility rate increases because of the increased mobility of nodes. The increased mobility could lead to less reliable connections between nodes and therefore the number of packets will decrease.

Next, we observe the effects of changing maximum mobility rate with respect to the average end-to-end delay. The results are captured in **Figure. 4.4**.

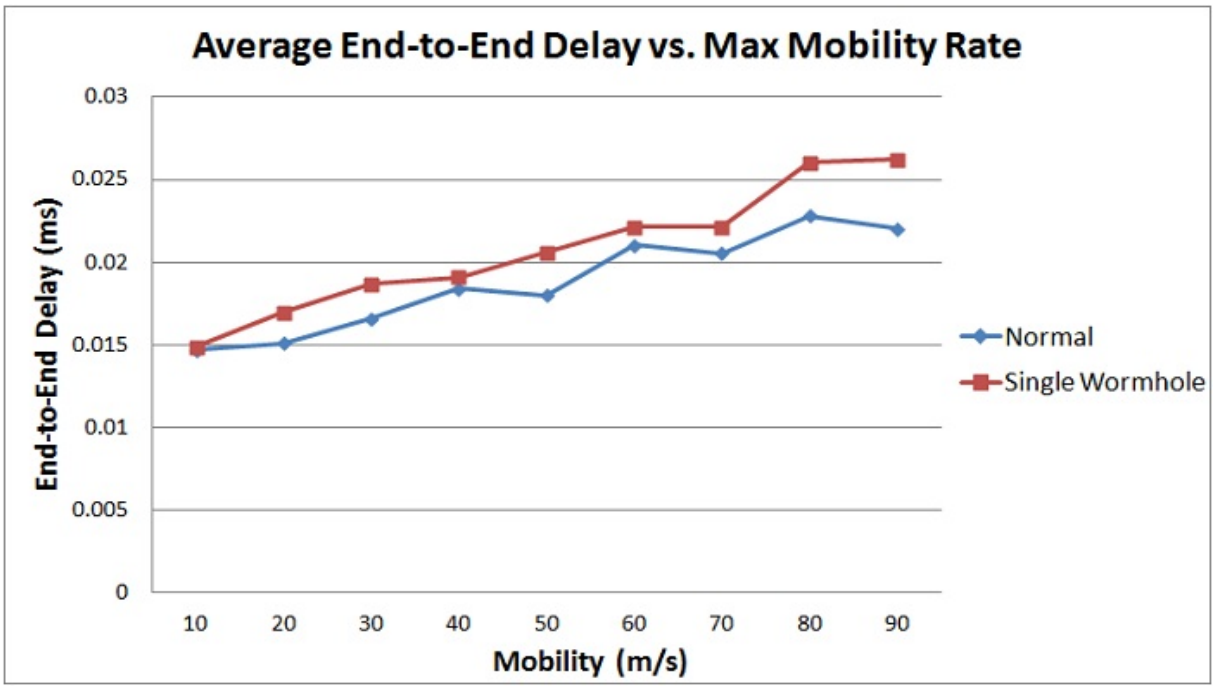

Figure 4.4: Method 2 - Average End-to-End Delay vs. Max Mobility

In **Figure. 4.4**, it can be observed that the average end-to-end delay increases when the network experiences a single wormhole attack throughout all mobility rates. Both methods have increasing average end-to-end delay because the speed at which the nodes are moving is also increasing, thus creating an environment where wireless communications need more time to get connections. Therefore, it is reasonable for average end-to-end delay to increase since more time is needed to create a stable connection to the destination.

4.2 HSAM vs. E-HSAM vs. E-HSAM-AES

4.2.1 Assumptions and Scope of the Simulations

The movement and direction of each node is randomized by the simulator and each node is restricted to move within a specified terrain of the MANET. All statistical data are recorded for all trials. The simulations are run 3 times for each trial to ensure the accuracy of the acquired data.

4.2.2 Simulation Parameters

Table. 4.2 outlines the main parameters for HSAM, E-HSAM and E-HSAM-AES:

Table 4.2: HSAM, E-HSAM and E-HSAM-AES Parameters

Routing Protocol	AODV
Traffic Type	CBR
MAC Protocol	IEEE 802.11
Mobility Model	Random Way Point
Number of Nodes	40
Default Wireless Power (txPower_dBm)	15 dBm
Malicious Wireless Power (txPower_dBm)	40 dBm
Max Movement Speed	10-90 m/s with 10 m/s increments
Simulation Terrain (m x m)	1000 x 1000
Simulation Time	150 Minutes
Packet Size	2048 Bytes (512 and 1024 can be used)

4.2.3 Performance Metrics

The following performance metrics are considered:

- Number of packets received by the destination node
- The packet delivery ratio (i.e. packets delivered to destination/total number of packets originated from source)
- The number of broken links: This is the number of previously valid routes which have become unavailable

4.2.4 Simulation Scenarios

In this section, we perform two different simulation scenarios. The first scenario is designed to compare HSAM and E-HSAM. The second scenario is to compare all three schemes, HSAM, E-HSAM and E-HSAM-AES.

The first scenario (so-called scenario 1) simulates a wormhole attack which targets the integrity of data packets which are traversing through the MANET. We chose nodes 1 and

30 to be the source and destination nodes respectively. Nodes 2 and 22 are chosen to be the malicious (mal) nodes which are implemented specifically to compromise the integrity of the data packets. The malicious nodes are chosen to be within a reasonable range of the source and destination nodes to ensure that some of the data packets will go through them. For the non-control implementation, if packets are compromised, a similar mechanism to the sending of RERR control packets is used to alert the source node that the route should no longer be used (hence, assuming it should be similar to a broken link). Consequently, the mechanism which sends RERR packets will also increment the routing table sequence numbers thus choosing the next available route. Both mal nodes have increased radio ranges (txPower_dBm of 40) which is 2.67 times greater than regular default node radio strength (txPower_dBm of 15).

The second scenario (so-called scenario 2) assumes the same nodes as the first. Once again, nodes 1 and 30 are chosen to be the source and destination nodes (respectively). Nodes 2 and 22 are chosen to be the malicious nodes, which are implemented specifically to compromise the integrity of the data packets. However, in this scenario, we assume that the malicious nodes are able to target the acknowledgement frames and are able to turn a 'Confidentiality Lost' field into an 'ACK' field. Also, if the message has been tampered, the destination node will consider the transfer incomplete since the packet received may not be the same as the one sent originally.

4.2.5 HSAM vs. E-HSAM

In this section, we implement E-HSAM and compare it against HSAM under scenario 1. As discussed earlier, E-HSAM replaces the data chunks from the original method with mock packets containing no original data from the real payload. Moreover, as an attempt to increase efficiency, we modify how the sender is notified of discovery of malicious routes. HSAM uses a similar procedure used in AODV to send notifications to the sender of a message. Consequently, we treat a wormhole link as a broken link and at the same time, we

update our blacklist. Moreover, the HSAM method resends all the data packet chunks again with a new route whereas E-HSAM continues the next mock packet chunk with another route obtained from the routing tables, thereby increasing efficiency and reducing redundancy. First, we examine the number of packets received by the destination node as well as the packet delivery ratio. The results are captured in **Figure. 4.5** and **Figure. 4.6** respectively.

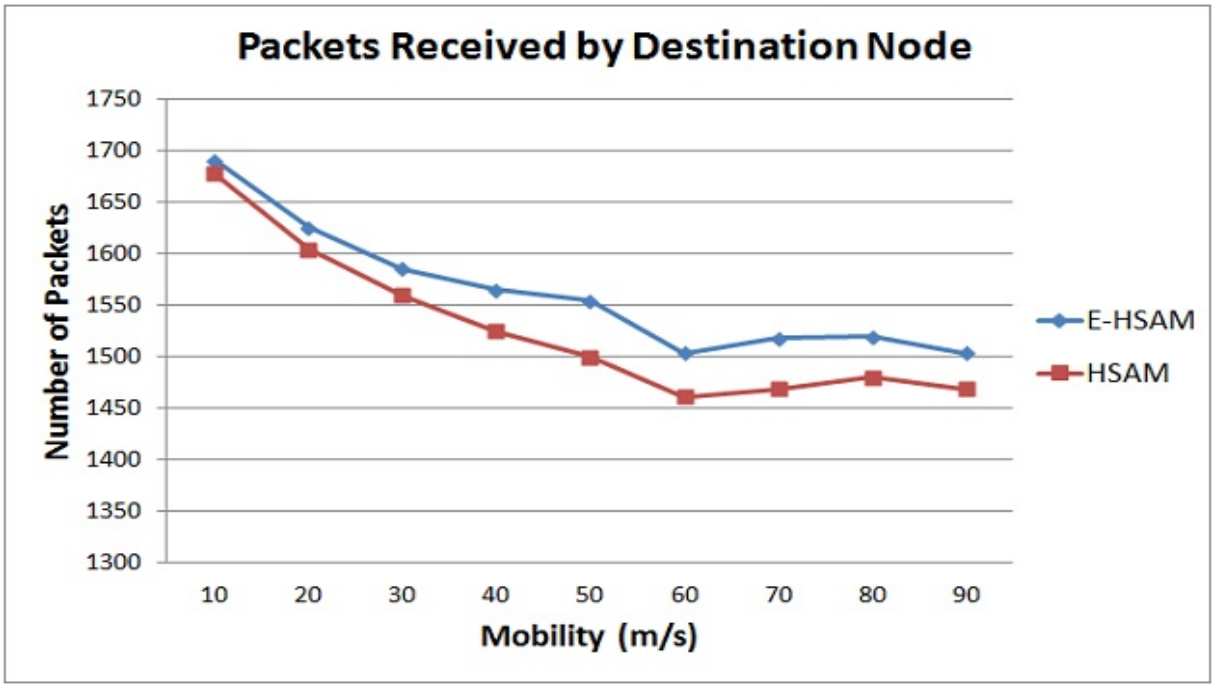

Figure 4.5: Packets Received vs. Max Mobility (E-HSAM)

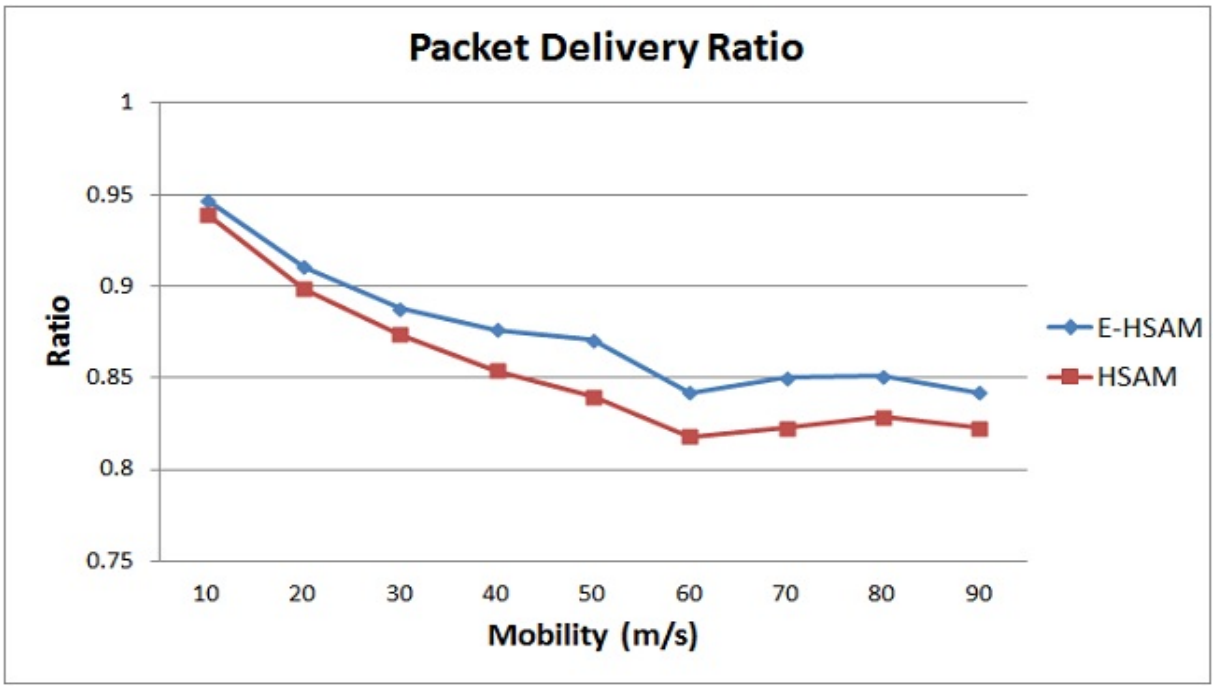

Figure 4.6: Packet Delivery Ratio vs. Max Mobility (E-HSAM)

In **Figure. 4.5**, it is observed that the number of packets which the destination receives is higher in E-HSAM compared to HSAM. This is understandable since E-HSAM automatically re-route the packets using the next available route and mock packet chunks are not dropped completely. On the other hand, HSAM drops the data packet chunks if there is a discrepancy with the data packet (i.e. if the hash value is no longer valid). A similar observation is cascaded in **Figure. 4.6** where E-HSAM is shown to have a higher packet delivery ratio compared to HSAM. As expected, the number of packets delivered as well as the packet delivery ratio steadily decreases as the mobility rate increases. This might be due to the fact that as nodes move around the simulated terrain with increased speed, connections have greater chance of failing since nodes can be out of range at any given time.

Next, we examine the number of broken links in both E-HSAM and HSAM. The results are captured in **Figure. 4.7**.

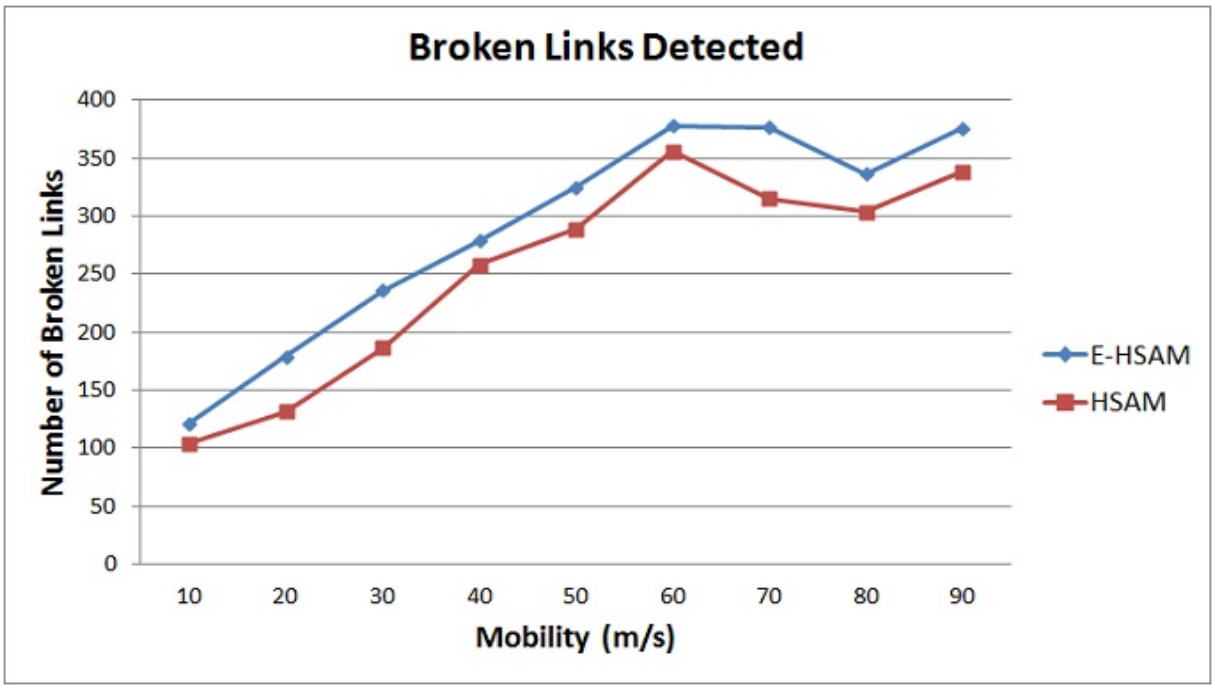

Figure 4.7: Broken Links vs. Max Mobility (E-HSAM)

In **Figure. 4.7**, it can be observed that the number of broken links is more prominent in E-HSAM compared to HSAM. This was expected since E-HSAM is based on the RERR mechanism of AODV where broken links are reported.

4.2.6 HSAM vs. E-HSAM vs. E-HSAM-AES

We will first examine the number of packets received as well as the delivery ratio when the node speed increases under scenario 2. The results are depicted in **Figure. 4.8** and **Figure. 4.9** respectively.

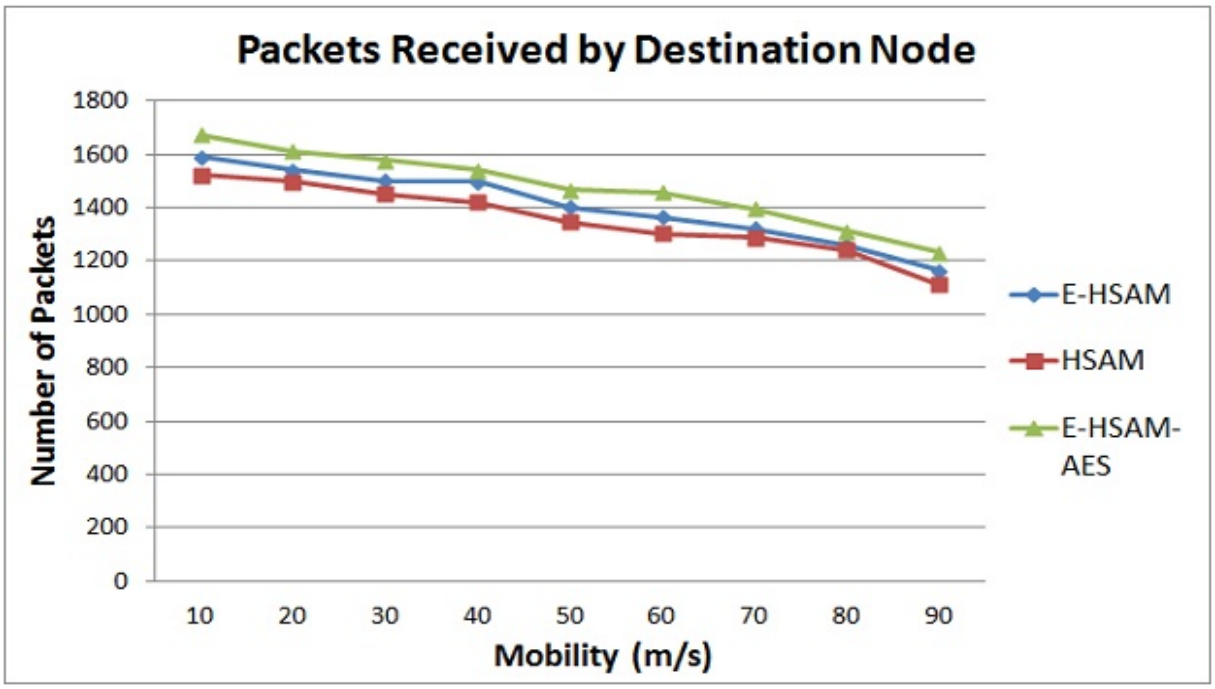

Figure 4.8: Packets Received vs. Max Mobility (E-HSAM-AES)

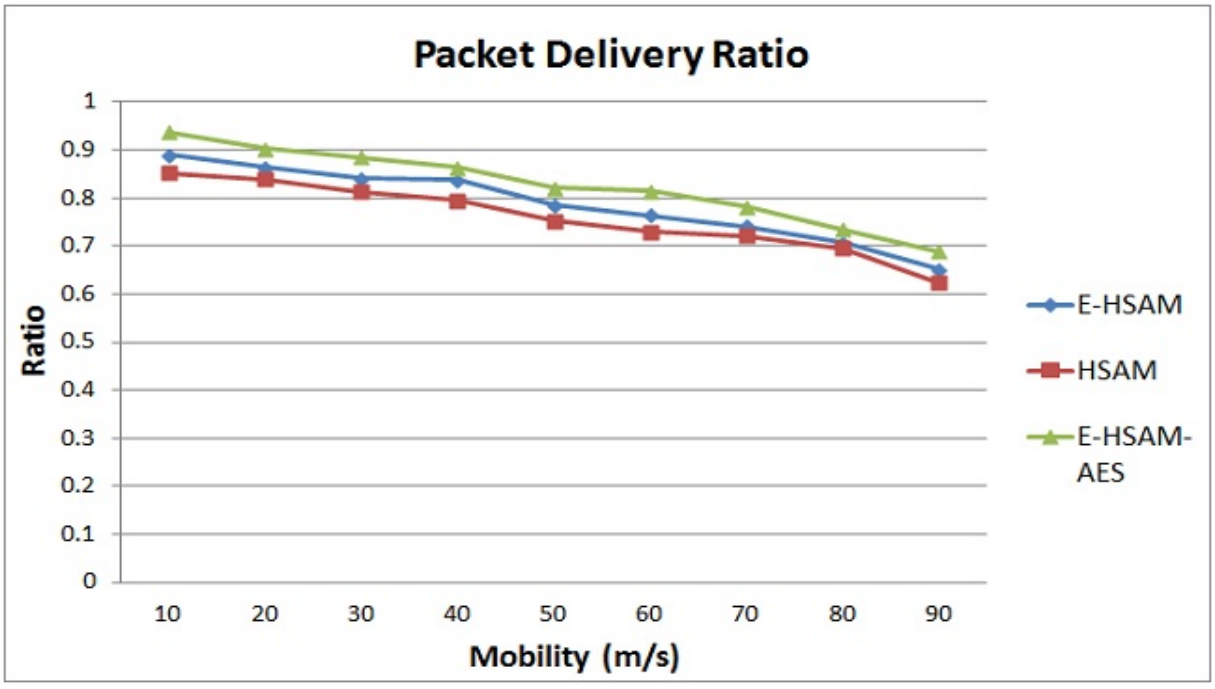

Figure 4.9: Delivery Ratio vs. Max Mobility (E-HSAM-AES)

In **Figure. 4.8**, it can be observed that for the three methods, there are minor differences in

the number of packets received at the destination node. Indeed, in E-HSAM-AES, if the data packet has been tampered with, the destination node will not consider a successful transfer. It is also observed that E-HSAM outperforms HSAM. The packet delivery ratio (**Figure. 4.9**) reflects on the number of successful packet transfers to the destination node. As with the number of packets received, the differences between all three nodes is not significant.

We now examine the number of broken links when the node speed increases. The results are depicted in **Figure. 4.10**.

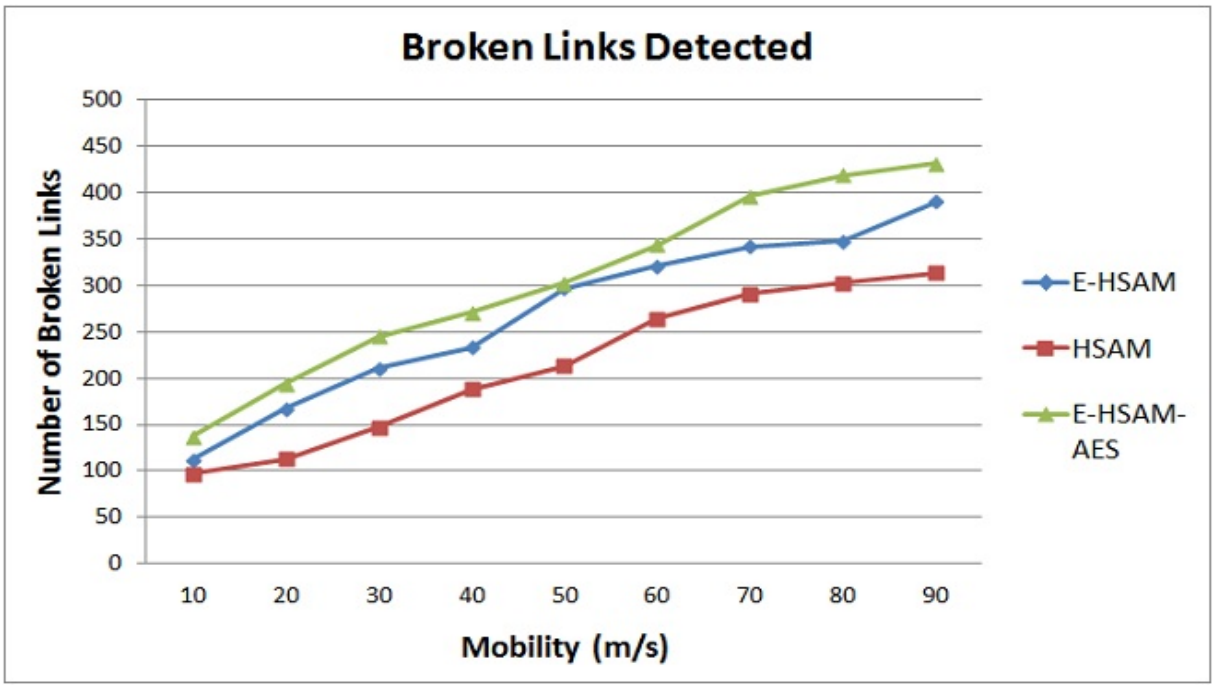

Figure 4.10: Broken Links vs. Max Mobility (E-HSAM-AES)

In **Figure. 4.10**, it can be observed that E-HSAM-AES leads the other two methods in the number of broken links experienced by the schemes. As mentioned previously, AES does indeed create more overhead for the data packet. Further, more computation is needed in order to encrypt and decrypt the packet. Therefore, it is expected that E-HSAM-AES will experience a higher number of broken links since packets could wander away from each other. As mobility increases, the difference in the number of broken links become more apparent due to the computational needs and greater overhead experienced by E-HSAM-AES, both

which can contribute to greater delay in transfers.

From the above observations, we can conclude that E-HSAM-AES is an effective scheme to help preventing against collaborative wormhole attacks. A combination of cryptography and hash markers makes copying and tampering attacks more difficult for malicious nodes. Although there is a slight increase in overhead to accommodate the encrypted packets, the additional security for data packets experienced in E-HSAM-AES is desirable and practical.

4.3 AODV-WADR-AES vs. AODV-WADR-TDES

4.3.1 Assumptions and Scope of the Simulations

We assume that the nodes in the simulation are normal functioning mobile devices and are part of the MANET. The movement and direction of each node is randomized by the simulator. Moreover, each node is restricted to move within the specified terrain of the MANET. We are using a frequently updated AES and TDES implementation called PolarSSL AES and PolarSSL TDES [26, 27]. The simulations are run 3 times for each trial to ensure the accuracy of the acquired data. Finally, all tests are performed in a software environment only.

4.3.2 Simulation Parameters

We first replicate the AODV-WADR-AES algorithm using the same parameters as in [6]. Those parameters are captured in **Table. 4.3**:

Table 4.3: AODV-WADR-AES Parameters

Routing Protocol	AODV
Traffic Type	CBR
Transport Protocol	TCP and UDP
MAC Protocol	IEEE 802.11
Mobility Model	Random Way Point
Number of Nodes	10, 25, 35, 50 and 60
Default Wireless Power (txPower_dBm)	15 dBm
Malicious Wireless Power (txPower_dBm)	40 dBm
Max Movement Speed	2 m/s
Cryptographic Algorithms	PolarSSL AES and TDES
Simulation Terrain (m x m)	1000 x 1000
Simulation Time	1000 seconds
Packet Size	2048 Bytes (512 and 1024 can be used)

The experiment has 2 different simulations for each metric measurement as well as each encryption algorithm. The 2 trials consist of:

- 1000 x 1000 terrain in TCP transport mode
- 1000 x 1000 terrain in UDP transport mode

4.3.3 Performance Metrics

In order to measure how effective one cryptographic protocol is against the other, we consider the end-to-end delay between the source and the destination nodes.

4.3.4 Simulation Scenarios

We simulate a wormhole attack which drops data packets which are traversing through the MANET. The two normal nodes which are trying to communicate with each other are nodes 1 and 8. Conversely, there will be two malicious nodes (nodes 6 and 10) involved in disrupting the normal communications between nodes 1 and 8. During route request, any control packets traversing through node 6 will be forwarded to node 10. Similarly, any control

packets traversing through node 10 will be forwarded to node 6. Conversely, data packets being routed through node 6 or 10 will be dropped. Both malicious nodes will have increased radio ranges (txPower_dBm of 40) which is 2.67 times greater than the regular default node radio strength (txPower_dBm of 15). Both the Transmission Control Protocol (TCP) and the User Datagram Protocol (UDP) are used since they are both popular protocols used in everyday communications such as email and internet browsing. The same simulation variables are used for both AODV-WADR-AES and AODV-WADR-TDES. Added to both algorithms are hash markers to the data packet in order to increase the data integrity.

4.3.5 Results on AODV-WADR-AES vs. AODV-WADR-TDES

We begin our analysis by evaluating the end-to-end delay of both the AES and TDES implementations. The results are depicted in **Figure. 4.11** and **Figure. 4.12** respectively when the terrain dimension is 1000m x 1000m.

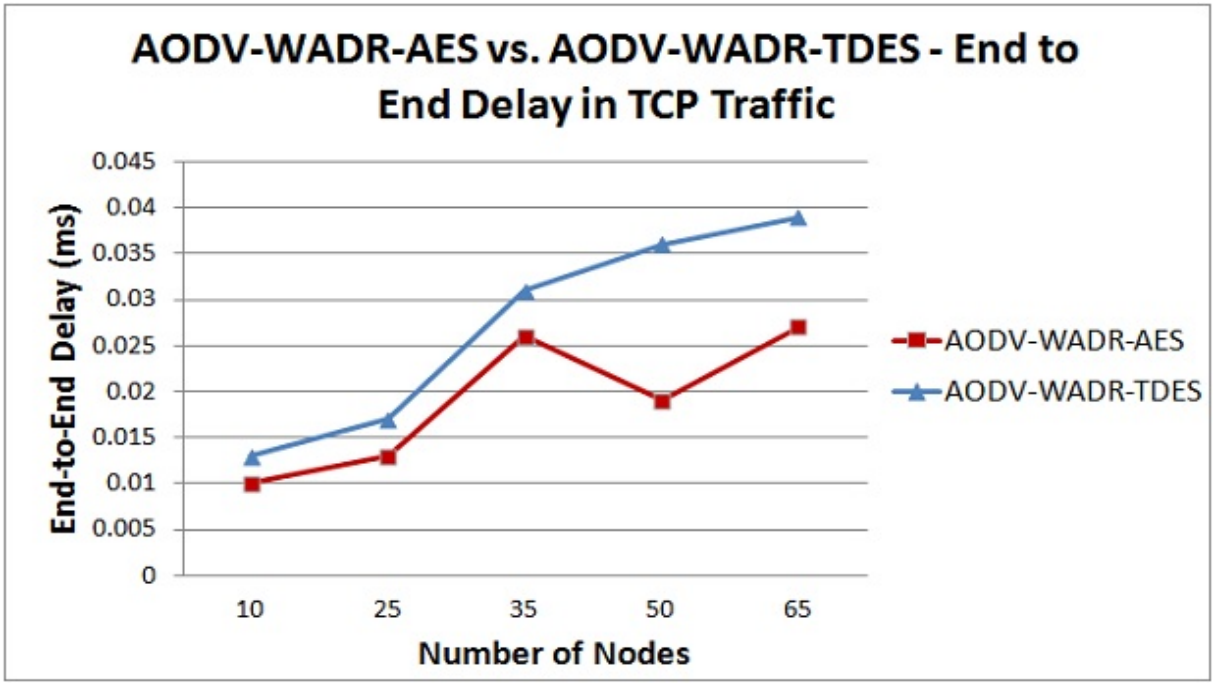

Figure 4.11: End-to-End Delay using TCP (AODV-WADR-AES)

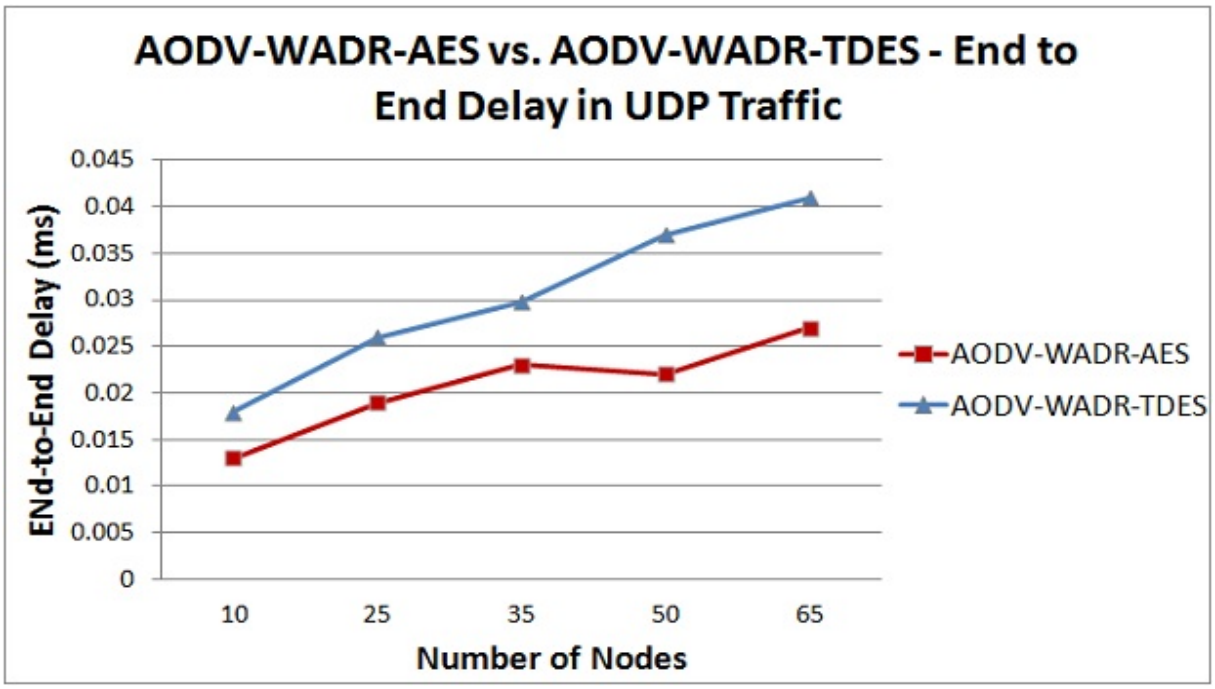

Figure 4.12: End-to-End Delay using UDP (AODV-WADR-AES)

In **Figures. 4.11 and 4.12**, it can be observed that the end-to-end delay for the TDES is higher than that of the AES implementation. The experiments performed by Wang and Hu [4] revealed that TDES was up to approximately three times slower than AES, which concurs with our observations. However, as pointed out in [21], DES was originally created to run more efficiently with hardware than software. Therefore, it is possible that if our tests were performed exclusively using hardware, the results may have been different. Nevertheless, the actual hardware tests are beyond the scope of this thesis, which explores the implementations through software.

4.4 A Timed and Secure Monitoring Implementation

4.4.1 Assumptions and Scope of the Simulations

The movement and direction of each node is randomized by the simulator. Moreover, each node is restricted to move within the specified terrain of the MANET. All three schemes, E-

HSAM, AODV-WADR-AES and the Timed and Secure Monitoring Implementation (TSMI), are tested against each other in this experiment. All statistical data are recorded for all trials. The simulations are run 3 times for each method to ensure the accuracy of the acquired data.

4.4.2 Simulation Parameters

Table. 4.4 describes the settings used by E-HSAM, AODV-WADR-AES and TSMI:

Table 4.4: TSMI Parameters

Routing Protocol	AODV
Traffic Type	CBR
MAC Protocol	IEEE 802.11
Mobility Model	Random Way Point
Number of Nodes	40
Default Wireless Power (txPower_dBm)	15 dBm
Malicious Wireless Power (txPower_dBm)	40 dBm
Max Movement Speed	10-90 m/s with 10 m/s increments
Cryptographic Algorithms	PolarSSL AES and TDES
Simulation Terrain (m x m)	1000 x 1000
Simulation Time	1000 seconds
Packet Size	2048 Bytes

4.4.3 Performance Metrics

The following performance metrics are considered:

- Number of packets received by destination node
- Packet delivery ratio (packets received by destination/total number of packets originating from source)
- Number of broken links

4.4.4 Simulation Scenarios

For all three methods, we simulate wormhole attacks which drop data packets which are traversing through the MANET. The two normal nodes which are trying to communicate with each other are nodes 1 and 30. There are two malicious nodes (nodes 5 and 20) involved in disrupting the normal communications between nodes 1 and 30. During route request, any control packets traversing through node 5 will be forwarded to node 20. Similarly, any control packets traversing through node 20 will be forwarded to node 5. Both malicious nodes have an increased radio range (txPower_dBm of 40) which is 2.67 times greater than regular default node radio strength (txPower_dBm of 15). Finally, data packets being routed through nodes 5 or 20 will be dropped.

4.4.5 Results using TSMI

We first analyze the number of packets received by the destination node and the packet delivery ratio. The results are captured in **Figure. 4.13 and 4.14** respectively.

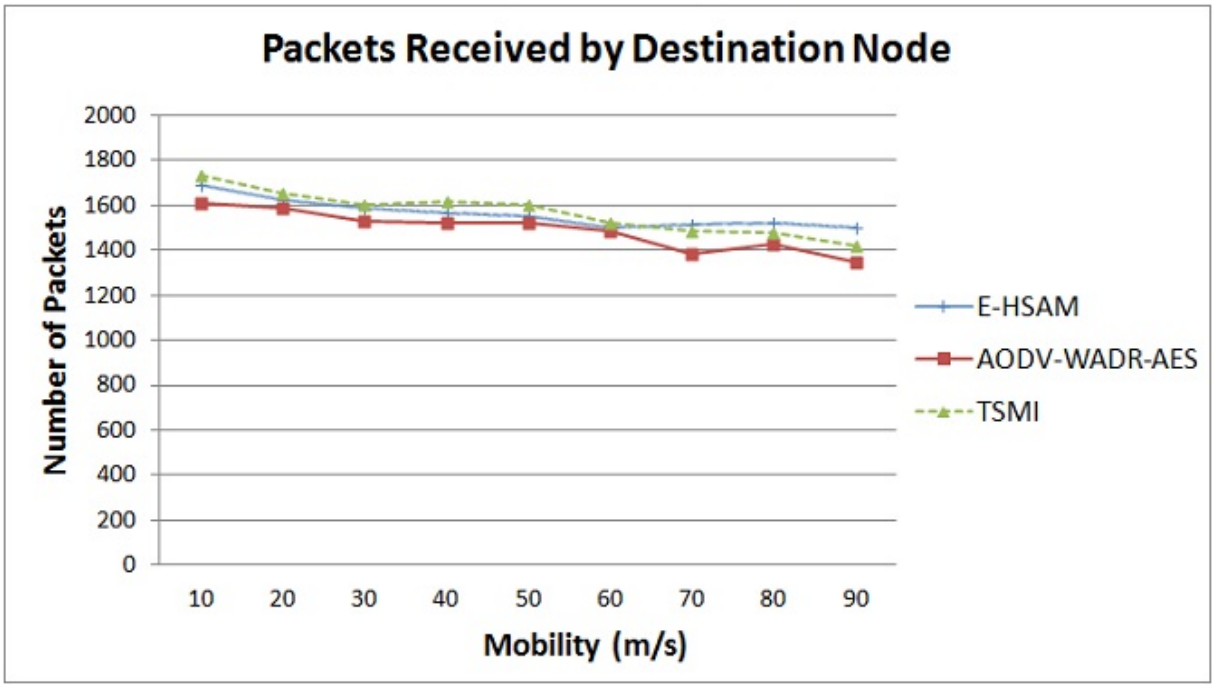

Figure 4.13: Number of packets received by destination node vs. Max Mobility (TSMI)

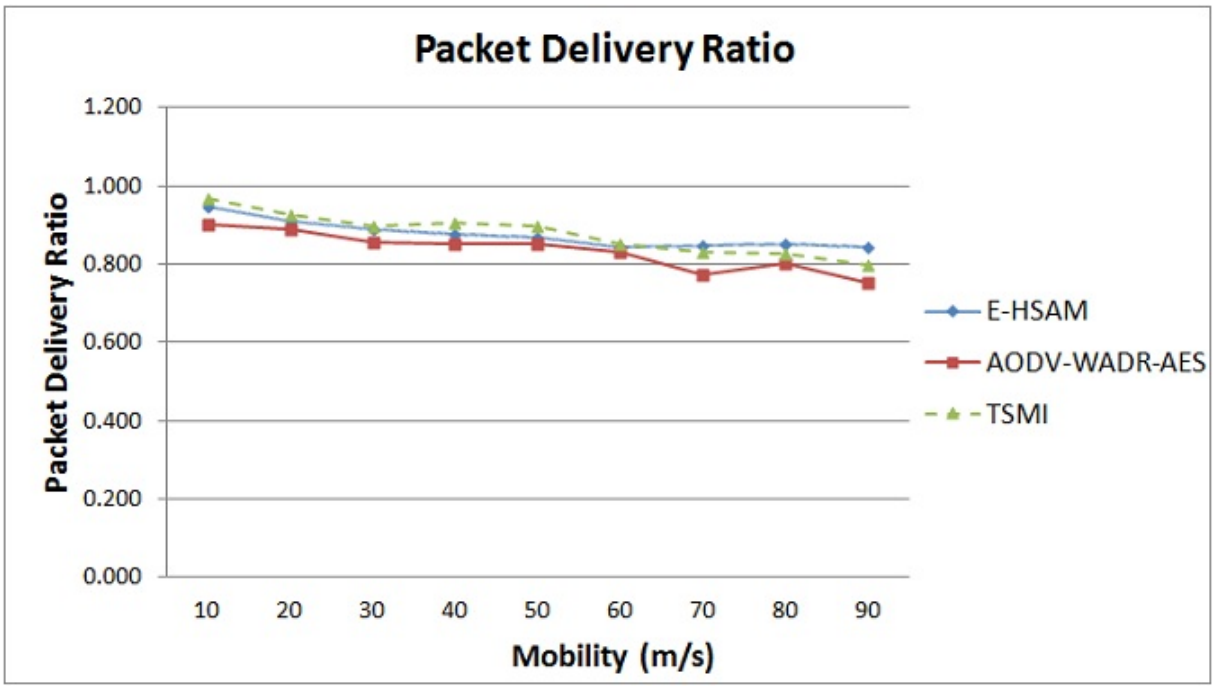

Figure 4.14: Packet Delivery Ratio vs. Max Mobility (TSMI)

In **Figure. 4.13 and 4.14**, it can be observed that the number of packets received is in general higher in TSMI compared to AODV-WADR-AES and E-HSAM. In TSMI, the average increase of packets received by the destination is 0.16% when compared with the E-HSAM scheme. Similarly, we observed an average increase of 4.86% when TSMI is compared to AODV-WADR-AES. Concurrently, the aforementioned results are also observed when comparing the packet delivery ratio of all three methods. All three methods yield similar results with TSMI having more packets delivered and a higher packet delivery ratio when the nodes have a maximum mobility speed between 10m/s to 60m/s. Within 10m/s and 60m/s, TSMI achieved 2.05% more packets than E-HSAM and 4.76% more packets when compared to AODV-WADR-AES. However, the number of packets delivered is lower in TSMI when the mobility speed is beyond 60m/s. This could be explained by the fact that TSMI also runs an encryption algorithm, whose timing can impact the results.

Next, we examine the number of broken links in each algorithm. The results are depicted in **Figure. 4.15**.

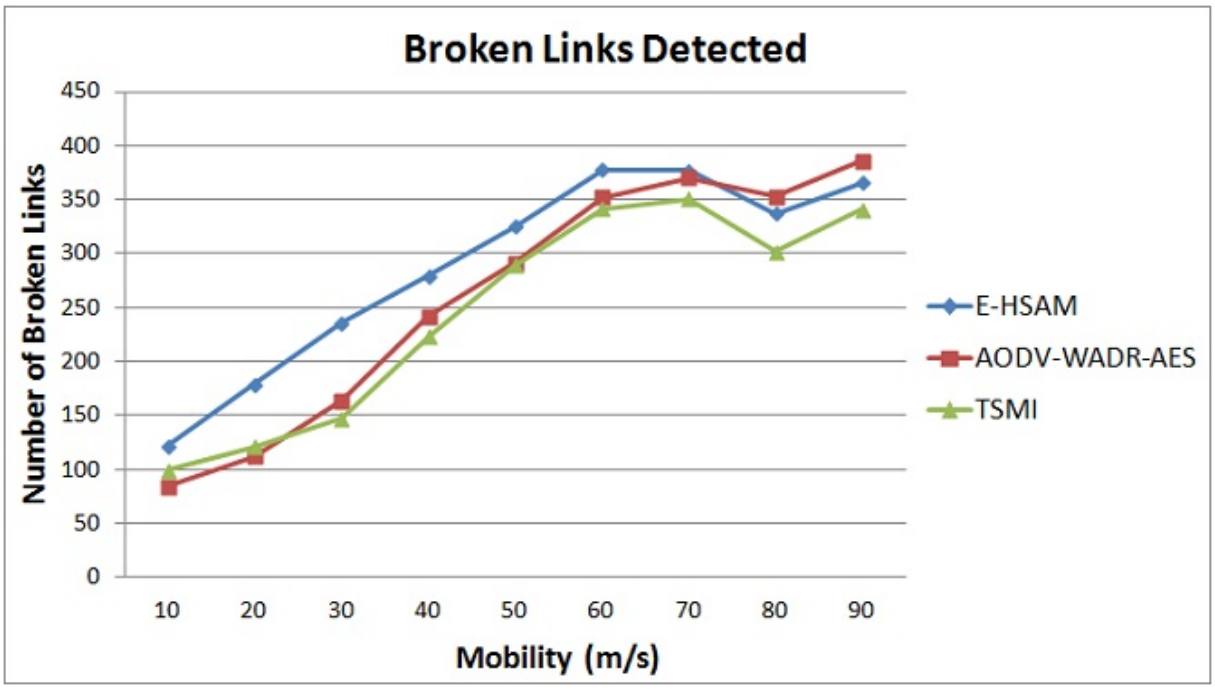

Figure 4.15: Number of Broken Links vs. Max Mobility (TSMI)

In **Figure. 4.15**, it can be observed that the number of broken links is smaller in TSMI compared to E-HSAM. This can be explained by the fact that in TSMI, if the route is 3-hops and the ATT is lower than MTT, the network will execute the normal AODV behavior whereas in E-HSAM the number of hops is greater or less than 3. Moreover, E-HSAM uses a method similar to the default AODV method to report RERR packets and broken links. This mechanism will discard the suspicious route and automatically increment the routing table sequence number before choosing the next route.

Finally, **Table. 4.5** depicts the number of data packets which are routed through the wormhole for all three methods.

Table 4.5: TSMI - Data Packets Routed Through Wormhole

Implementation	Data packets routed by wormhole
E-HSAM	0
AODV-WADR-AES	0
TSMI	0

From **Table. 4.5**, it is obvious that all three methods were effective against wormhole attacks since no packets were routing through them.

Chapter 5

Conclusions

In this thesis, we have: (1) enhanced HSAM with E-HSAM which improves the security of HSAM as well as its performance; (2) implemented AODV-WADR-TDES as a complement to AODV-WADR-AES in order to observe the performance differences within MANETs; (3) designed a novel scheme TSMI which implements the features of E-HSAM and AODV-WADR-AES in order to create a highly secure method which prevents collaborative wormhole attacks in MANETs.

Addressing the potential security weaknesses of HSAM leads to the design of E-HSAM. By replacing the actual data packets with mock packet chunks, E-HSAM can safely analyze the routes without worrying about data chunks being copied. Moreover, we are able to increase the efficiency of the HSAM and make the algorithm adaptable to collaborative wormhole attacks. Further, we also compared E-HSAM-AES with E-HSAM and HSAM.

By analyzing the idea of AODV-WADR-AES, we are able to discover the strengths and weaknesses of the encryption methodology used therein. Moreover, we compared the effectiveness of AODV-WADR-AES against AODV-WADR-TDES which uses the TDES algorithm instead of AES. Although AODV-WADR-AES yields impressive results, it is only effective if the route chosen is 3-hops between the source and destination nodes. Moreover, the implementation of AODV-WADR-AES may not be able to outperform AODV-WADR-

TDES in a hardware environment since TDES performs much better in hardware devices.

Combining the features of E-HSAM and AODV-WADR-AES, allowed us to design a new scheme called TSMI. TSMI is able to address the potential weaknesses of E-HSAM and AODV-WADR-AES. The results for TSMI demonstrate its effectiveness in preventing collaborative wormhole attacks. However, there are limitations to our scheme where the effectiveness begins to decline after 60m/s. Our method is useful and applicable to MANETs because it is improbable for mobile nodes to move at 60m/s. Moreover, the trade-off between performance and security outweigh these limitations.

We have a few ideas which pertain to the future work of research in the area of MANETs and collaborative wormhole attacks. Firstly, in our implementation of AODV-WADR-AES, we chose to use Cipher Block Chaining (CBC) as our main mode of operation. There are other modes which are also applicable for use. Such modes include the Cipher Feedback (CFB), Counter (CTR) and Output Feedback (OFB) modes. However, we do emphasize that ECB should not be used because of its simplicity and the use of the same key to encrypt individual blocks of plaintext.

Another implementation could be to go a little further with the security of data packets and encrypt them as they traverse through the MANET. Since AES is an efficient cryptographic algorithm [28], the overall performance and overhead for sending the data packet could be optimized. However, we emphasize that data encryption is a "double edged sword" and could compromise network performance. Conversely, if the potential performance reduction is minimal and the data packet is more secure, then we believe it is an acceptable trade-off. Although we evaluated AES in comparison with TDES, we did not use TDES as the main encryption algorithm in TSMI for the scope of this thesis. For future experiments, TDES can be used in TSMI in order to examine its performance and security. Additionally, we can use TSMI-AES and TSMI-TDES with hardware devices to evaluate the performance of each. We postulate that the TDES implementation could perform better than the AES solution since DES and TDES perform much better in hardware devices [28].

In our experiments, as mentioned in Chapter 4, we have tested the AODV protocol in MANETs using unicast methods. For later experiments, we may expand our ideas using multicast methods. We postulate that adverse effects on performance of TSMI in multicast would be minimal (as seen with our results in unicast).

Bibliography

[1] R. Jhaveri, A. Patel, J. Parmar, and B. Shah, "MANET Routing Protocols and Wormhole Attack against AODV," *International Journal of Computer Science and Network Security*, vol. 10, pp. 12–18, April 2010.

[2] V. Mahajan, M. Natu, and A. Sethi, "Analysis of wormhole intrusion attacks in manets," in *IEEE Military Communications Conference (MILCOM 2008)*, San Diego, California, USA, pp. 1 –7, November 17-19, 2008.

[3] F. Nait-Abdesselam, "Detecting and avoiding wormhole attacks in wireless ad hoc networks," *IEEE Communications Magazine*, vol. 46, pp. 127 –133, April 2008.

[4] Y.-C. Hu, A. Perrig, and D. Johnson, "Packet leashes: a defense against wormhole attacks in wireless networks," in *the 22nd Annual Joint Conference of the IEEE Computer and Communications (INFOCOM 2003)*, San Francisco, California, USA, vol.3, pp. 1976 – 1986, April 1-3, 2003.

[5] G. Mamatha and D. S. C. Sharma, "A Highly Secured Approach against Attacks in MANETS," *International Journal of Computer Theory and Engineering*, vol. 2, pp. 815–819, October 2010.

[6] E. A. Panaousis, L. Nazaryan, and C. Politis, "Securing aodv against wormhole attacks in emergency manet multimedia communications," in *Proceedings of the 5th International ICST Mobile Multimedia Communications Conference (Mobimedia 2009)*, Brussels, Belgium, Belgium, pp. 34:1–34:7, 2009.

[7] I. Woungang, S. Dhurandher, and V. Koo, "Preventing packet dropping and message tampering attacks on aodv-based mobile ad hoc network," in *Proceedings of the International Conference on Computer, Information and Telecommunication Systems (CITS 2012)*, Amman, Jordan. May 14-16, 2012.

[8] C. H. Vu and A. Soneye, *"Collaborative Attacks on MANETs: An Analysis of Collaborative Attacks on Mobile Ad hoc Networks"*. LAP Lambert Academic Publishing, Germany, 2010.

[9] R. Maheshwari, J. Gao, and S. R. Das, "Detecting wormhole attacks in wireless networks using connectivity information," in *26th IEEE International Conference on Computer Communications (INFOCOM 2007)*, Anchorage, Alaska, USA, pp. 107–115, May 14-16, 2007.

[10] A. Mtibaa and F. Kamoun, "Mmdv: Multipath and mpr based aodv routing protocol," in *the IFIP 5th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net 2006)*, Lipari, Sicily, Italy, pp. 137–144, June 14-17, 2006.

[11] H. Xu, X. Wu, H. Sadjadpour, and J. Garcia-Luna-Aceves, "A unified analysis of routing protocols in manets," *IEEE Transactions on Communications*, vol. 58, pp. 911 –922, March 2010.

[12] N. Meghanathan, "A manet multicast routing protocol for stable trees based on the inverse of link expiration times," in *IEEE Consumer Communications and Networking Conference (CCNC 2012)*, Jackson, MS, USA, pp. 947 –951, January 14-17, 2012.

[13] M. Hellman, "An overview of public key cryptography," *IEEE Communications Magazine*, vol. 40, pp. 42 –49, May 2002.

[14] U. N. I. of Standards and Technology, "Federal Information Processing Standards Publication 197," tech. rep., US Department of Commerce / National Bureau of Standards, Springfield, Virginia. November 26, 2001.

[15] I. Khalil, S. Bagchi, and N. Shroff, "Liteworp: a lightweight countermeasure for the wormhole attack in multihop wireless networks," in *International Conference on Dependable Systems and Networks (DSN 2005)*, West Lafayette, Indiana, USA, pp. 612 – 621, June 28 - July 1, 2005.

[16] J. Eriksson, S. V. Krishnamurthy, and M. Faloutsos, "Truelink: A practical countermeasure to the wormhole attack in wireless networks," in *International Conference on Network Protocols (ICNP 2008)*, Orlando, Florida, USA, pp. 75–84, October 19-22, 2006.

[17] A. Singh and K. Vaisla, "A mechanism for detecting wormhole attacks on wireless ad hoc network," *International Journal of Computer Science and Network Security*, vol. 2, pp. 27–31, September 2010.

[18] S. Choi, D. young Kim, D.-H. Lee, and J.-I. Jung, "Wap: Wormhole attack prevention algorithm in mobile ad hoc networks," in *Proceedings of IEEE International Conference on Sensor Networks, Ubiquitous and Trustworthy Computing (SUTC 2008)*, Taichung, Taiwan, pp. 343 –348, June 11-13, 2008.

[19] T. Hayajneh, P. Krishnamurthy, and D. Tipper, "Deworm: A simple protocol to detect wormhole attacks in wireless ad hoc networks," in *Proceedings of the 3rd International Conference on Network and System Security (NSS 2009)*, Gold Coast, Queensland, Australia, pp. 73–80, October 19-21, 2009.

[20] S. K. Dhurandher, I. Woungang, A. Gupta, and B. K. Bhargava, "E2siw: An energy efficient scheme immune to wormhole attacks in wireless ad hoc networks," in *International Conference on Advanced Information Networking and Applications Workshops*, Fukuoka, Japan, pp. 472–477, March 26-29, 2012.

[21] H. O. Alanazi, B. B. Zaidan, A. A. Zaidan, H. A. Jalab, M. Shabbir, and Y. Al-Nabhani, "New comparative study between des, 3des and aes within nine factors," *Computer Research Repository*, vol. abs/1003.4085, 2010.

[22] O. Gonzalez, M. Howarth, and G. Pavlou, "Detection of packet forwarding misbehavior in mobile ad-hoc networks," in *Proceedings of the 5th international conference on Wired/Wireless Internet Communications (WWIC 2007)*, Coimbra, Portugal, pp. 302–314, May 23-25, 2007.

[23] B. Preneel, "Aes and nessie: Cryptographic algorithms for the 21st century," Eurocacs 2002 Keynote. Budapest, Hungary. March 24-28, 2002, http://www.esat.kuleuven.ac.be/p̃reneel.

[24] U. N. I. of Standards and T. S. P. (SP), "Recommendation for the triple data algorithm," Tech. Rep. FIPS SP 800-67, US Department of Commerce / National Bureau of Standards, Springfield, Virginia. May 2004.

[25] S. Lucks and R. Weis, "How to make des-based smartcards fit for the 21-st century: Cryptographic techniques for advanced security requirements," in *Proceedings of the 4th working conference on smart card research and advanced applications on Smart card research and advanced applications*, Bristol, United Kingdom, pp. 93–114, September 20-22, 2000.

[26] I. Woungang, S. Dhurandher, and V. Koo, "Comparison of two security protocols for preventing packet dropping and message tampering attacks on aodv-based mobile ad hoc networks," in *submission to the IEEE 4th International Workshop on Mobility Management in the Networks of the Future World (MobiWorld 2012)*, Anaheim, California, USA. December 3-7, 2012.

[27] "Polarssl aes and tdes," retrieved July 28, 2012, http://polarssl.org/.

[28] Y. Wang and M. Hu, "Timing evaluation of the known cryptographic algorithms," in *International Conference on Computational Intelligence and Security (CIS 2009)*, Beijing, China, vol. 2, pp. 233 –237, December 11-14, 2009.