TRUST BASED DATA SECURITY AND SECURE ANONYMOUS ROUTING FOR MANETS

A PROJECT REPORT

Submitted by

SHALINI INFANTA A

VINOTHINI R

in partial fulfilment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

LOYOLA-ICAM

COLLEGE OF ENGINEERING AND TECHNOLOGY

CHENNAI – 600 034

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2017

ANNA UNIVERSITY : CHENNAI 600 025

BONAFIDE CERTIFICATE

Certified that this project report **"TRUST BASED DATA SECURITY AND SECURE ANONYMOUS ROUTING FOR MANETS"** is the bonafide work of **"SHALINI INFANTA A (311113104048), VINOTHINI R (311113104054)"** who carried out the project work under my supervision.

SIGNATURE	**SIGNATURE**
Dr. K Gopalakrishnan	Ms. Misiriya Shahul Hameed
HEAD OF DEPARTMENT	**SUPERVISOR**
	Assistant Professor
Department of Computer Science and Engineering Loyola-ICAM College of Engineering and Technology Chennai – 600 034	Department of Computer Science and Engineering Loyola-ICAM College of Engineering and Technology Chennai – 600 034

Submitted for the project viva voce held on _____

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

First of all, we praise God, the almighty, for providing us this opportunity and granting us the capability to proceed successfully. Working on this project has been a rewarding experience. We would like to express our sincere thanks and gratitude to our Director, **Rev. Dr. Ignacy Arockyaa SJ** and our Dean**, Rev. Dr. John Pragasam SJ** for their continuous support and encouragement. We thank them for making the right resources available at the right time.

We would like to extend our gratitude to our Principal, **Dr. Jose Swaminathan** for his aspiring guidance and invaluably constructive criticism during the project work.

We would like to express our sincere thanks to the head of the department, **Dr. K Gopalakrishnan** for his brilliant comments and suggestions.

We also extend our gratitude to our esteemed project guide**, Ms. Misiriya Shahul Hameed, M.E.,** for providing valuable insights and resources leading to the successful completion of our project. Without her supervision and constant help this project would not have been possible.

We would also like to thank the project coordinator, **Mr. L Remegius Praveen Sahayaraj, M.Tech.,** and all the faculty members of our department for their critical advice and guidance which was helpful in completion of our project.

Last but not the least we place a deep sense of gratitude to our family members and our friends who have been a constant source of inspiration during the preparation of this project work.

ABSTRACT

For many applications of the Mobile Ad hoc Networks (MANETs) deployed in adversary environments anonymous communications with data confidentiality and integrity is important. The existing protocols are vulnerable to security threats like global eavesdropper attacks, fake routing packets attacks or denial-of-service (DoS) broad-casting attacks, traffic analysis attacks and insider attacks. Authenticated Anonymous Secure Routing protocol (AASR) defends the attacks and provides sufficient anonymity with the help of group signature and key encrypted onion routing. In this work a unified trust management scheme has been integrated with AASR protocol in order to enhance the routing and data security in MANETs. Simulation results have demonstrated the effectiveness of the proposed protocol with improved performance in terms of throughput, packet received ratio, packet loss ratio and delay when compared to the existing ones.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATIONS

MANET	Mobile Ad-hoc Network
AASR	Authenticated Anonymous Secure Routing
DSDV	Destination Sequenced Distance Vector Routing
AODV	Ad-hoc On-demand Distance Vector
DSR	Dynamic Source Routing
DoS	Denial od Service
RREQ	Route REQuest
RREP	Route REPly
NS	Network Simulator
NAM	Network AniMator
TCP	Tranmission Control Protocol
UDP	User Datagram Protocol
FTP	File Transfer Protocol
LL	Logic Link
MAC	Media Access Control
TCL	Tool Command Language
SG	Sink Gateway
CIC	Cluster In-Charge
SIC	Sector In-Charge

LIST OF TABLES

CHAPTER 1

INTRODUCTION

Mobile ad hoc networks (MANETs) represent complex distributed systems that consist of wireless mobile nodes that can dynamically and freely self-organize into arbitrary and temporary ad hoc network topologies. This allows people and devices to seamlessly internetwork in areas where no pre-existing communication infrastructure exists, for example disaster recovery environments. The unique characteristics of MANETs, such as dynamic topology and resource constraint devices, pose a number of nontrivial challenges for efficient and lightweight security protocols design.

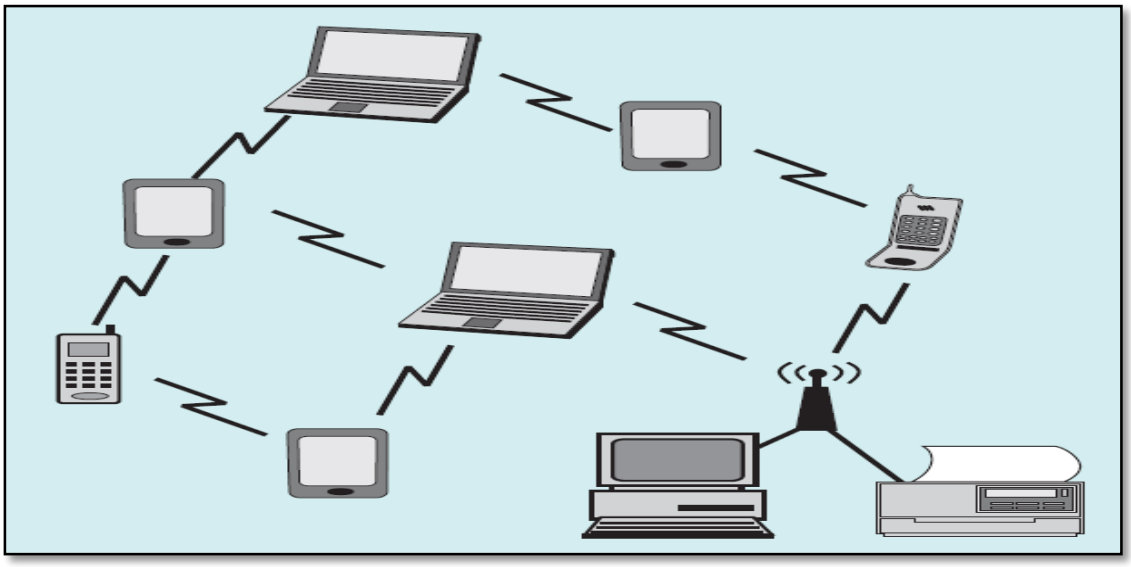

Figure 1.1: Mobile Ad hoc Network

In Figure 1.1, each node act as both host and router. That is it is autonomous in behavior. Multi-hop radio relaying- When a source node and destination node for a message is out of the radio range, the MANETs are capable of multi-hop routing. Distributed nature of operation for security, routing and host configuration. A centralized firewall is absent here. The nodes can join or leave the network anytime, making the network topology dynamic in nature. Mobile nodes are characterized with less memory, power and light weight features.

1.1 PROBLEM STATEMENT

The distinctive features of mobile ad hoc networks (MANETs), like dynamic topology, open wireless transmission medium, nomadic and distributed nature and lack of centralized infrastructure of security protection may lead to many security vulnerabilities in MANET routing. The secure routing protocols provide reliable data link layer and route maintenance support, but are not designed to cope with malicious disruptions and need enhanced security to establish secure routing paths for data transmission and also to reduce the packet delay as well as to be more active in detecting link failures, caused either by the mobility or adversary attacks.

1.2 MANET SECURITY

In recent years mobile ad hoc networks (MANETs) have received tremendous attention because of their self-configuration and self-maintenance capabilities. While early research effort assumed a friendly and cooperative environment and focused on problems such as wireless channel access and multi hop routing, security has become a primary concern in order to provide protected communication between nodes in a potentially hostile environment. Although security has long been an active research topic in wire line networks, the unique characteristics of MANETs present a new set of nontrivial challenges to security design. These challenges include open network architecture, shared wireless medium, stringent resource constraints, and highly dynamic network topology. Consequently, the existing security solutions for wired networks do not directly apply to the MANET domain.

The ultimate goal of the security solutions for MANETs is to provide security services, such as authentication, confidentiality, integrity, anonymity, and availability, to mobile users. In order to achieve this goal, the security solution should provide complete protection spanning the entire protocol stack. The security issues in each layer. In this article we consider a fundamental

security problem in MANET: the protection of its basic functionality to deliver data bits from one node to another. In other words, we seek to protect the network connectivity between mobile nodes over potentially multi hop wireless channels, which is the basis to support any network security services. Multi hop connectivity is provided in MANETs through two steps:

- Ensuring one-hop connectivity through link-layer protocols (e.g., wireless medium access control, MAC)
- Extending connectivity to multiple hops through network layer routing and data forwarding protocols (e.g., ad hoc routing).

One distinguishing characteristic of MANETs from the security design perspective is the lack of a clear line of defense. Unlike wired networks that have dedicated routers, each mobile node in an ad hoc network may function as a router and forward packets for other peer nodes. The wireless channel is accessible to both legitimate network users and malicious attackers. There is no well-defined place where traffic monitoring or access control mechanisms can be deployed. As a result, the boundary that separates the inside network from the outside world becomes blurred. On the other hand, the existing ad hoc routing protocols, such as Ad Hoc On Demand Distance Vector (AODV) and Dynamic Source Routing (DSR) and wireless MAC protocols, such as 802.11 typically assume a trusted and cooperative environment. As a result, a malicious attacker can readily become a router and disrupt network operations by intentionally disobeying the protocol specifications.

Security never comes for free. When more security features are introduced into the network, in parallel with the enhanced security strength is the ever-increasing computation, communication, and management overhead. Consequently, network performance, in terms of scalability, service availability, robustness, and so on of the security solutions, becomes an important concern in a resource-constrained ad hoc network. While many contemporary proposals

focus on the security vigor of their solutions from the cryptographic standpoint, they leave the network performance aspect largely unaddressed. In fact, both dimensions of security strength and network performance are equally important, and achieving a good trade-off between two extremes is one fundamental challenge in security design for MANETs.

1.3 SECURITY ATTACKS

A MANET provides network connectivity between mobile nodes over potentially multi hop wireless channels mainly through link-layer protocols that ensure one-hop connectivity, and network-layer protocols that extend the connectivity to multiple hops. These distributed protocols typically assume that all nodes are cooperative in the coordination process. This assumption is unfortunately not true in a hostile environment. Because cooperation is assumed but not enforced in MANETs, malicious attackers can easily disrupt network operations by violating protocol specifications. The main network-layer operations in MANETs are ad hoc routing and data packet forwarding, which interact with each other and fulfill the functionality of delivering packets from the source to the destination. The ad hoc routing protocols exchange routing messages between nodes and maintain routing states at each node accordingly. Based on the routing states, data packets are forwarded by intermediate nodes along an established route to the destination. Nevertheless, both routing and packet forwarding operations are vulnerable to malicious attacks, leading to various types of malfunction in the network layer. While a comprehensive enumeration of the attacks is out of our scope, such network-layer vulnerabilities generally fall into one of two categories: routing attacks and packet forwarding attacks, based on the target operation of the attacks.

Recent research efforts have also identified the vulnerabilities of the link-layer protocols, especially the de facto standard IEEE 802.11 MAC protocol, for MANETs. It is well known that 802.11 WEP is vulnerable to several types of

cryptography attacks due to the misuse of the cryptographic primitives. The 802.11 protocol is also vulnerable to DoS attacks targeting its channel contention and reservation schemes. The attacker may exploit its binary exponential back off scheme to deny access to the wireless channel from its local neighbors. Because the last winner is always favored among local contending nodes, a continuously transmitting node can always capture the channel and cause other nodes to back off endlessly. Moreover, back offs at the link layer can incur a chain reaction in upper layer protocols using back off schemes (e.g., TCP's window management). Another vulnerability of 802.11 comes from the NAV field carried in the request to send/clear to send (RTS/CTS) frames, which indicates the duration of channel reservation. An adversarial neighbor of either the sender or the receiver may overhear the NAV information and then intentionally introduce a 1-bit error into the victim's link-layer frame by wireless interference. The corrupted frame has to be discarded by the receiver after error detection. This effectively constitutes another type of DoS attack.

The MANETS set new challenges for network security and the need of an hour is to pay more attention to the security threats posed on the network. Following are the concerned issues in security of ad hoc networks:

- **Nodes Acting as Routers**: As nodes themselves are participating in relaying of messages, any malicious node in the network can easily misuse the message traffic either by dropping messages or by generating false messages etc.
- **Limited Resources**: Due to the limitation of network resources in mobile ad hoc networks, the various cryptographic solutions applicable to wired networks are not directly applicable. Therefore there is a need for new security solutions which can find their application in this challenging domain.

- **Mobility of Nodes**: Dynamically changing network topology results in more opportunities for the malicious nodes to attack.
- **Location of Nodes**: Since Ad hoc networks are formed for a purpose, the deployment environment may not be very security sensitive. For Example, the nodes deployed in the battle field or in the forests for tracking wild animals etc. may invite many security threats and attacks.
- **Wireless Medium**: Interoperability is very easy in a wireless medium. Therefore, there is a lack of privacy and the important messages can be eavesdropped and modified easily

1.3.1 CATEGORIZING NETWORK ATTACKS

Attacks on the ad hoc networks can be broadly categorized as Passive Attacks and Active Attacks.

- **Passive Attacks** - The main aim of passive attackers is to steal the valuable information from the targeted networks. Attackers do not disturb the normal network functioning like inducing false packets or dropping packets. They simply become a part of the network but continuously keeps an eye on the network traffic thus in turn violating the message confidentiality constraint. Since they do not initiate any malicious activity to disrupt the normal functioning of the network, it becomes very difficult to identify such attacks. Examples of such types of attacks are traffic analysis, traffic monitoring and eavesdropping.
- **Active Attacks** - Active attackers tamper with the network traffic like cause congestion, propagation of incorrect routing information etc. Due to their active participation, their detection and prevention can be done using suitable prevention algorithms. Examples of passive attacks include modification attack, impersonation, fabrication and message replay. Attacks can also be classified depending upon the position of the attacker in the network.

- **External attacks -** External Attacks are the attacks made by the unauthorized nodes which are not a part of the Network. External attackers can flood bogus packets in the network, impersonation etc. Their aim can be to cause congestion or to disrupt normal network functioning.
- **Internal attacks -** Internal Attacks are caused by the authorized nodes in the network. The reason for their malicious behavior may be the following: Hijacking those (authorized) nodes by some external attacker and then using them for launching internal attacks in the network. Selfishness to save their limited resources like battery power, processing capabilities, and the communication bandwidth and exploiting other nodes for their benefit.

1.3.2 ROUTING ATTACKS

Flooding Attack

It is the basic form of Denial of Service (DoS). The aim of this attack is to paralyze the whole network by exhausting network resources like bandwidth of the network, battery of nodes. Radio jamming and battery exhaustion methods are the tools to conduct this attack in the network. It can be caused in some of the following ways:

- Attackers may initiate massive bogus route request (RREQ) packets that will definitely be rebroadcast on and on by other nodes. Bogus may be in the sense that the destination address does not exist in the network. As there will not be any reply for these RREQs, network will be flooded leading to the consumption of battery power and bandwidth of all nodes. For example, consider a simple network scenario shown in Figure 1.2. Here node D generates RREQ packets destined to the node address H, which is actually not present in the network and broadcast it to all neighboring nodes(C, G and E) in the network. Since no neighbor node

will be able to find H, they will again rebroadcast it assuming that some other nodes may be able to find the path to 1. H. In this way battery power and bandwidth are being wasted without doing any useful work with RREQ flooding.

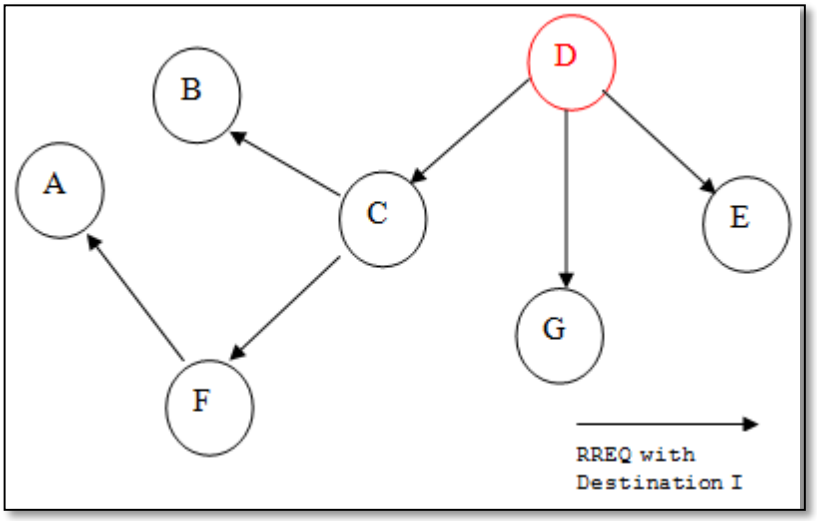

Figure 1.2: Example of Flooding Attack

- Analogous to RREQ flooding, a malicious node can do data flooding also. In this technique after setting path to all the nodes, attacker node sends useless data packets to them.
- **Detection of flooding attack can be done in following ways**:

The detection of any attack can be performed with the cooperation of genuine nodes in the network. For detecting the presence of a malicious node responsible for RREQ flooding in the network, rate of packet (or RREQ) generation of any node should be checked by the neighboring nodes. If the rate exceeds some threshold value (set either statically or dynamically by the algorithm) that node should be put into the blacklist and this information should be broadcasted in the network.

Similarly for the prevention of data flooding, a threshold for data rate generation by any node in the network is to be set and should be checked periodically for all the neighboring nodes in the network.

1.3.3 SLEEP DEPRIVATION ATTACK

Sleep deprivation attack is a type of flooding attack where either a specific node or a group of nodes is targeted whose resources need to be exhausted. This attack can be implemented by forcing the targeted node to use its vital resources e.g. battery, network bandwidth and computing power by sending false requests for existent or non-existent destination nodes. In the meantime it cannot process the requests coming from genuine nodes. The main aim of the malicious node is to minimize the genuine nodes lifetime by wasting its valuable resources. As a result the victim node is not able to participate in routing mechanisms and become unreachable by other nodes in the network.

As an example, consider the network scenario in Figure 1.3 where a malicious node C is exhausting the resources of node D by sending bogus data packets or bogus RREQ packets for processing. Some of the proposed solutions to the sleep deprivation attack are:

- A clustering based prevention method is proposed by Sarkar et al. in which suggest the formation of clusters in the networks as in least cluster change algorithm. It proposes that the node with the lowest node identifier number is assigned the cluster head. The cluster head is updated whenever two cluster heads come in direct contact. A cluster head should forward packets for a particular source-destination pair in its cluster until a threshold value (say 10 packets) is reached. After that the cluster head breaks its connection with that node. In this way, it results in preventing a node from sending excessive traffic.

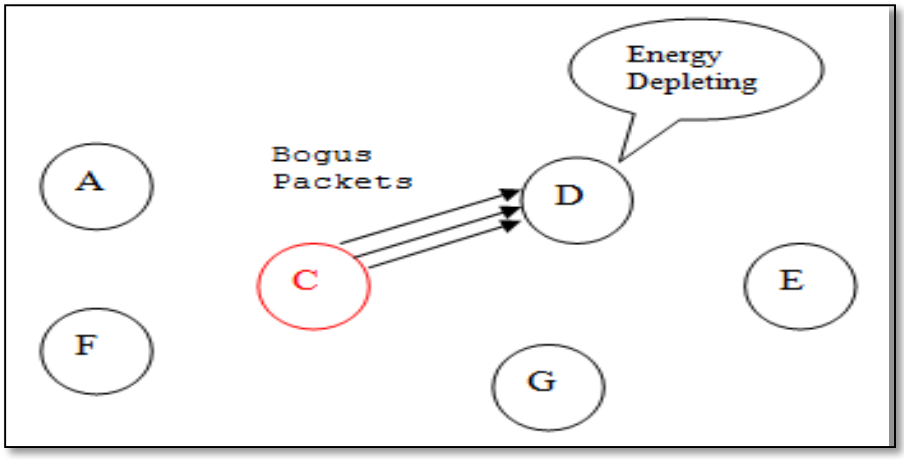

Figure 1.3 Example of Sleep Deprivation Attack

- Another solution as proposed by uses a hierarchy based model for the detection of sleep deprivation attacks in sensor networks. All sensor nodes in the network are arranged in a hierarchy of Sink gateway (SG), Cluster In-charge (CIC) having maximum energy level and maximum degree of connectivity in the cluster, Sector Monitor which is nearest Neighbor of the CIC having maximum detective capability for an anomaly, Sector In-charge (SIC) having maximum energy level among all neighbors of CIC and collects sensing data from a sector) and Leaf nodes (LN) having capability to sense data.
- The whole network is logically divided into clusters, headed by CIC and clusters are further divided into sectors headed by SIC. Data collection request is initiated by the CIC and sent to the SIC which forwards this request to its associated LNs. LNs in turn returns the sensed data to SIC which forwards the collected data to the SM.
- SM checks for the validity or non-validity of the collected data and sends the packets marked as valid or non-valid to the CIC. CIC takes the final decision for preventing the rate of false positive

detection. Then it forwards valid data to the SG along with rejecting the non-valid data. Also suspected node gets added into the SG's isolation list for future prevention.

1.4 EXISTING ROUTING PROTOCOLS

In cryptographic functions, a trapdoor is a common concept that defines a one-way function between two sets. A global trapdoor is an information collection mechanism in which intermediate nodes may add information elements, such as node IDs, into the trapdoor. Only certain nodes, such as the source and destination nodes can unlock and retrieve the elements using pre-established secret keys. The usage of trapdoor requires an anonymous end-to end key agreement between the source and destination.

There are many anonymous on-demand routing protocols. Similar to the ad hoc routing, there are two categories: topology-based and location-based, or in other words, node identity centric and location centric .They compare the protocols in Table I, in terms of the key distribution assumption, node anonymity in route discovery, and packet authentication. Our observations are summarized as follows:

First of all, the routing protocols are designed to work in different scenarios. AO2P, PRISM, and ALERT are designed for location-based or location-aided anonymous communications, which require localization services.

Since ours is for general MANETs, they focus on the topology-based routing rather than location-based routing. Secondly, as mentioned in Section I, SDAR, Anon DSR, MASK, and D-ANODR have problems in meeting the unindentifiability and unlinkability. The node IDs in a neighbor- hood and along a route are possibly exposed in SDAR and Anon DSR, respectively. The plain node IDs are used in the route request of MASK and D-ANODR. In this work, the node’s pseudonym is used instead of its real ID, to avoid the information leakage during RREQ and RREP processes.

Disadvantages

Many Anonymous routing protocols are available. But it does not detect the active attackers effectively.

- Low Throughput in the presence of adversaries.
- Heavy Packet Loss
- High delay

1.5 PROPOSED SYSTEM - TRUST MANAGEMENT SCHEME

With recent advances in wireless technologies and mobile devices, Mobile Ad hoc Networks (MANETs) have become popular as a key communication technology in military tactical environments such as establishment of communication networks used to coordinate military deployment among the soldiers, vehicles, and operational command centers. There are many risks in military environments needed to be considered seriously due to the distinctive features of MANETs, including open wireless transmission medium, nomadic and distributed nature, lack of centralized infrastructure of security protection. Therefore, security in tactical MANETs is a challenging research topic. Although some excellent work has been done on detection based approaches based on trust in MANETs, most of existing approaches do not exploit direct observation method to evaluate the trust of an observed node.

In this work, trust is interpreted as the degree of belief that a node performs as expected. Uncertainty in trust evaluation is also recognized. Based on this interpretation, a trust management scheme is developed to enhance the security of MANETs. The difference between this scheme and existing schemes is the use of uncertain reasoning to derive trust values. Uncertain reasoning was initially proposed from the artificial intelligence community to solve the problems in expert systems, which have frequent counter-factual results. The

elasticity and flexibility of uncertain reasoning make it successful in many fields, such as expert systems, multiagent systems, and data fusion. The contributions of this work are outlined as follows:

- A unified trust management scheme is developed that enhances the security in MANETs using uncertain reasoning. With direct observation from an observer node, the trust value is derived using Bayesian inference, which is a type of uncertain reasoning when the full probability model can be defined.
- The proposed scheme differentiates data packets and control packets, and meanwhile excludes the other causes that result in dropping packets, such as unreliable wireless connections and buffer overflows.
- We evaluate the proposed scheme by integrating it with a MANET routing protocol AASR, with the NS2 simulator. Extensive simulation results show the effectiveness of the proposed scheme. Throughput and packet delivery ratio can be improved significantly, with slightly increased average end-to-end delay and overhead of messages.

1.6 APPLICATIONS FOR MANETS

The field of wireless and mobile communications has experienced an unprecedented growth during the past decade. Current second-generation (2G) cellular systems have reached a high penetration rate, enabling worldwide mobile connectivity. Mobile users can use their cellular phone to check their email and browse the Internet. Recently, an increasing number of wireless local area network (LAN) hot spots is emerging, allowing travelers with portable computers to surf the Internet from airports, railways, hotels and other public locations. Broadband Internet access is driving wireless LAN solutions in the

home for sharing access between computers. In the meantime, 2G cellular networks are evolving to 3G, offering higher data rates, infotainment and location-based or personalized services.

However, all these networks are conventional wireless networks, conventional in the sense that as prerequisites, a fixed network infrastructure with centralized administration is required for their operation, potentially consuming a lot of time and money for set-up and maintenance. Furthermore, an increasing number of devices such as laptops, personal digital assistants (PDAs), pocket PCs, tablet PCs, smart phones, MP3 players, digital cameras, etc. are provided with short-range wireless interfaces. In addition, these devices are getting smaller, cheaper, more user friendly and more powerful. This evolution is driving a new alternative way for mobile communication, in which mobile devices form a self- creating, self-organizing and self-administering wireless network, called a mobile ad hoc network.

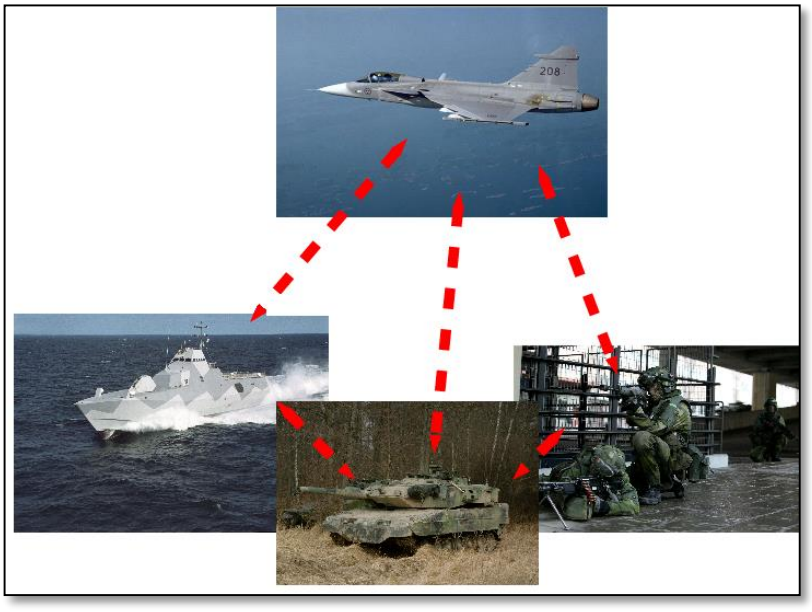

Figure 1.4: Applications of MANETS

CHAPTER 2

LITERATURE SURVEY

Introduction

In this section the findings of papers on Mobile Ad hoc Networking, Short Group Signatures and Trust Management Scheme has been discussed.

1. Short Group Signatures

Dan Boneh *et.al* (2004) proposed a short group signature scheme. Signatures in the scheme are approximately the size of a standard RSA signature with the same security. Security of the group signature is based on the Strong Diffie-Hellman assumption and a new assumption in bilinear groups called the Decision Linear assumption. The security of the system is proved by the random oracle model, using a variant of the security definition for group signatures recently given by Bellare, Micciancio, and Warinschi.

2. Mobile Ad hoc Networking (MANET) - Routing Protocol Performance Issues and Evaluation Considerations

S. Corson and J. Macker *et.al* (1999) proposed that with recent performance advancements in computer and wireless communications technologies, advanced mobile wireless computing is expected to see increasingly widespread use and application, much of which will involve the use of the Internet Protocol (IP) suite. The vision of mobile ad hoc networking is to support robust and efficient operation in mobile wireless networks by incorporating routing functionality into mobile nodes. Such networks are envisioned to have dynamic, sometimes rapidly-changing, random, multihop topologies which are likely composed of relatively bandwidth-constrained wireless links. Within the Internet community, routing support for mobile hosts is presently being formulated as "mobile IP" technology. This is a technology

to support nomadic host "roaming". The host may be directly physically connected to the fixed network on a foreign subnet, or be connected via a wireless link, dial-up line, etc. Supporting this form of host mobility (or nomadicity) requires address management, protocol interoperability enhancements and the like, but core network functions such as hop-by-hop routing still presently rely upon pre-existing routing protocols operating within the fixed network.

3. ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad-hoc Networks

Jiejun Kong *et.al* (2003) demonstrates that in hostile environments, the enemy can launch traffic analysis against interceptable routing information embedded in routing messages and data packets. Allowing adversaries to trace network routes and infer the motion pattern of nodes at the end of those routes may pose a serious threat to covert operations. We propose ANODR, an anonymous on-demand routing protocol for mobile ad hoc networks deployed in hostile environments. For route anonymity, ANODR prevents strong adversaries from tracing a packet flow back to its source or destination; for location privacy, ANODR ensures that adversaries cannot discover the real identities of local transmitters. The design of ANODR is based on “broadcast with trapdoor information”, a novel network security concept which includes features of two existing network and security mechanisms, namely “broadcast” and “trapdoor information”.

4. High Security for Manet Using Authentication and Intrusion Detection with Data Fusion

K.K.Lakshmi Narayanan, A.Fidal Castro *et.al* (2012) demonstrated that in Mobile Ad Hoc Network (MANET), Multimodal Biometric technology plays

a vital role in giving security between user-to-device authentications. This paper concentrates on the Intrusion Detection and authentication with data fusion in MANET. To overcome the fault in unimodal biometric systems, Multimodal biometrics is set out to work with Intrusion Detection Systems. Each and every device has dimensions and estimation limitations, many devices to be selected and with the help of Dempster-Shafter theory for data fusion observation precision gets increased. Based on the security posture, system concludes which biosensor (IDS) to select and whether user authentication (or IDS input) is essential. By every authentication device and Intrusion Detection System (IDS), the decisions are made in a fully distributed manner.

5. Ad hoc On-Demand Distance Vector (AODV) Routing

C. Perkins *et.al* (2003) proposed the Ad hoc On-Demand Distance Vector (AODV) routing protocol which is intended for use by mobile nodes in an ad hoc network. It offers quick adaptation to dynamic link conditions, low processing and memory overhead, low network utilization, and determines unicast routes to destinations within the ad hoc network. It uses destination sequence numbers to ensure loop freedom at all times (even in the face of anomalous delivery of routing control messages), avoiding problems (such as "counting to infinity") associated with classical distance vector protocols.

6. AnonDSR: Efficient Anonymous Dynamic Source Routing for Mobile Ad-Hoc Networks

Song. R. *et.al* (2005) proposed that security, anonymity, and scalability are still important issues for mobile ad hoc network routing protocols. We first expose the limitations of several existing mobile ad hoc network routing protocols with security and anonymity constraints and analyze their scalabilities. Based on the analysis, we propose a new anonymous dynamic

source routing protocol (AnonDSR) to provide three levels of security protection. Scalabilities with security constraints are compared and the new protocol is analyzed to show it has strong security and anonymity protection, and very good scalability.

7. USOR: An Unobservable Secure On-Demand Routing Protocol for Mobile Ad Hoc Networks

Zhiguo Wan, Kui Ren, and Ming Gu *et.al* (2012) proposed that Privacy-preserving routing is crucial for some ad hoc networks that require stronger privacy protection. A number of schemes have been proposed to protect privacy in ad hoc networks. However, none of these schemes offer complete unlinkability or unobservability property since data packets and control packets are still linkable and distinguishable in these schemes. In this paper, we define stronger privacy requirements regarding privacy-preserving routing in mobile ad hoc networks. Then we propose an unobservable secure routing scheme USOR to offer complete unlinkability and content unobservability for all types of packets. USOR is efficient as it uses a novel combination of group signature and ID-based encryption for route discovery. Security analysis demonstrates that USOR can well protect user privacy against both inside and outside attackers. We implement USOR on ns2, and evaluate its performance by comparing with AODV and MASK. The simulation results show that USOR not only has satisfactory performance compared to AODV, but also achieves stronger privacy protection than existing schemes like MASK.

8. A Hierarchical Identity Based Key Management Scheme in Tactical Mobile Ad Hoc Networks

F. Richard Yu, Helen Tang, Peter C. Mason, and Fei Wang *et.al* (2010) proposed that Hierarchical key management schemes would serve well for military applications where the organization of the network is already

hierarchical in nature. Most of the existing key management schemes concentrate only on network structures and key allocation algorithms, ignoring attributes of the nodes themselves. Due to the distributed and dynamic nature of MANETs, it is possible to show that there is a security benefit to be attained when the node states are considered in the process of constructing a private key generator (PKG). In this paper, we propose a distributed hierarchical key management scheme in which nodes can get their keys updated either from their parent nodes or a threshold of sibling nodes. The dynamic node selection process is formulated as a stochastic problem and the proposed scheme can select the best nodes to be used as PKGs from all available ones considering their security conditions and energy states. Simulation results show that the proposed scheme can decrease network compromising probability and increase network lifetime in tactical MANETs.

9. MASK: Anonymous On-Demand Routing in Mobile Ad Hoc Networks

Yanchao Zhang, Wei Liu, Wenjing Lou and Yuguang Fang *et.al* (2006) proposed that the shared wireless medium of mobile ad hoc networks facilitates passive, adversarial eavesdropping on data communications whereby adversaries can launch various devastating attacks on the target network. To thwart passive eavesdropping and the resulting attacks, we propose a novel anonymous on demand routing protocol, termed MASK, which can accomplish both MAC-layer and network-layer communications without disclosing real IDs of the participating nodes under a rather strong adversary model. MASK offers the anonymity of senders, receivers, and sender-receiver relationships in addition to node unlocatability and untrackability and end-to-end flow untraceability. It is also resistant to a wide range of attacks. Moreover, MASK preserves the high routing efficiency as compared to previous proposals. Detailed simulation studies have shown that MASK is highly effective and efficient.

10. Security Enhancements for Mobile Ad Hoc Networks with Trust Management Using Uncertain Reasoning

Zhexiong Wei, Helen Tang, F. Richard Yu, Maoyu Wang, and Peter Mason *et.al* (2014) proposed that the distinctive features of mobile ad hoc networks (MANETs), including dynamic topology and open wireless medium, may lead MANETs suffering from many security vulnerabilities. In the proposed trust management scheme, the trust model has two components: trust from direct observation and trust from indirect observation. With direct observation from an observer node, the trust value is derived using Bayesian inference, which is a type of uncertain reasoning when the full probability model can be defined. Indirect observation, is obtained from neighbour nodes of the observer node, the trust value is derived using the Dempster-Shafer theory, which is another type of uncertain reasoning when the proposition of interest can be derived by an indirect method. Combining these two components in the trust model, more accurate trust values of the observed nodes in MANETs can be obtained. Extensive simulation results show the effectiveness of the proposed scheme. Specifically, throughput and packet delivery ratio can be improved significantly with slightly increased average end to- end delay and overhead of messages.

Conclusion

Detailed study of the above papers conclude that integrating Trust Management Scheme with the existing AASR protocol increases the throughput and reduces the delay, packet loss ratio.

CHAPTER 3

IMPLEMENTATION

3.1 ANONYMOUS COMMUNICATION AND SECURITY

Anonymity is defined as the state of being unidentifiable within a set of subjects. In MANETs, the requirements of anonymous communications can be described as a combination of unindentifiability and unlinkability. The essence of anonymous communications is to hide the sender's and/or receiver's identities from outside observers. As a result, adversaries cannot correlate eavesdropped traffic information to actual network traffic patterns. The shared wireless medium of MANET's introduces opportunities for passive eavesdropping on data communications. Adversaries can easily overhear all the messages flying in the air without physically compromising a node. The key to implementing a secure communication for MANET is to develop appropriate anonymous secure routing protocols with end to end encryption for data traffic.

Onion Routing

In onion routing, instead of making socket connections directly to a responding machine, initiating applications make connections through a sequence of machines called onion routers. The onion routing network allows the connection between the initiator and responder to remain anonymous. Anonymous connections hide who is connected to whom, and for what purpose, from both outside eavesdroppers and compromised onion routers. If the initiator also wants to remain anonymous to the responder, then all identifying information must be removed from the data stream before being sent over the anonymous connection. Onion routers in the network are connected by longstanding (permanent) socket connections. Anonymous connections through the network are multiplexed over the longstanding connections. For any

anonymous connection, the sequence of onion routers in a route is strictly defined at connection setup. However, each onion router can only identify the previous and next hop along a route. Data passed along the anonymous connection appear different at each onion router, so data cannot be tracked en route, and compromised onion routers cannot cooperate by correlating the data stream each sees. We will also see that they cannot make use of replayed onions or replayed data.

The onion routing network is accessed via a series of proxies. An initiating application makes a socket connection to an application proxy. This proxy messages connection message format (and later data) to a generic form that can be passed through the onion routing network. It then connects to an onion proxy, which defines a route through the onion routing network by constructing a layered data structure called an onion. The onion is passed to the entry funnel that occupies one of the longstanding connections to an onion router and multiplexes connections to the onion routing network at that onion router. That onion router will be the one for whom the outermost layer of the onion is intended. Each layer of the onion defines the next hop in a route. The figure 3.1 shows a Single Onion Layer

An onion router that receives an onion peels off its layer, identifies the next hop, and sends the embedded onion to that onion router. The last onion router forward data to an exit funnel, whose job is to pass data between the onion routing network and the responder. In addition to carrying next-hop information, each onion layer contains key seed material from which keys are generated for crypting data sent forward or backward along the anonymous connection. (We define forward to be the direction in which the onion travels and backward as the opposite direction.)

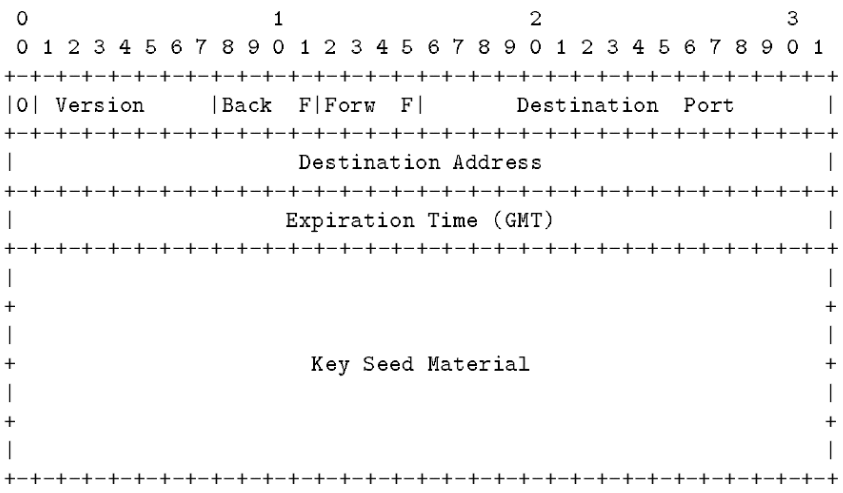

Figure 3.1: A Single Onion Layer

Once the anonymous connection is established, it can carry data. Before sending data over an anonymous connection, the onion proxy adds a layer of encryption for each onion router in the route. As data move through the anonymous connection, each onion router removes one layer of encryption, so it arrives at the responder as plaintext. This layering occurs in the reverse order for data moving back to the initiator. Therefore data that have passed backward through the anonymous connection must be repeatedly post-crypted to obtain the plaintext. By layering cryptographic operations in this way, we gain an advantage over link encryption. As data move through the network it appears different to each onion router. Therefore, an anonymous connection is as strong as its strongest link, and even one honest node is enough to maintain the privacy of the route. In link encrypted systems, compromised nodes can trivially cooperate to uncover route information. Onion routers keep track of received onions until they expire. Replayed or expired onions are not forwarded, so they cannot be used to uncover route information, either by outsiders or compromised onion routers. Note that clock skew between onion routers can only cause an onion router to reject a fresh onion or to keep track of processed

onions longer than necessary. Also, since data are encrypted using stream ciphers, replayed data will look different each time it passes through a properly operating onion router.

Although we call this system onion routing, the routing that occurs here does so at the application layer of the protocol stack and not at the IP layer. More specifically, we rely upon IP routing to route data passed through the longstanding socket connections. An anonymous connection is comprised of portions of several linked longstanding multiplexed socket connections. Therefore, although the series of onion routers in an anonymous connection is fixed for the lifetime of that anonymous connection, the route that data actually travels between individual onion routers is determined by the underlying IP network. Thus, onion routing may be compared to lose source routing. Onion routing depends upon connection-based services that deliver data uncorrupted and in order. This simplifies the specification of the system. TCP socket connections, which are layered on top of a connectionless service like IP, provide these guarantees.

Group Signature

Group signature scheme can provide authentications without disturbing the anonymity. Every member in a group may have a pair of group public and private keys issued by the group trust authority (i.e., group manager). The member can generate its own signature by its own private key, and such signature can be verified by other members in the group without revealing the signer's identity. Only the group trust authority can trace the signer's identity and revoke the group keys. Any member of the group can sign messages, but the resulting signature keeps the identity of the signer secret. In some systems there is a third party that can trace the signature, or undo its anonymity, using a special trapdoor. Some systems support revocation where group membership

can be selectively disabled without affecting the signing ability of unrevoked members.

Secure Packet Forwarding

The protection of routing message exchange is only part of the network-layer security solution for MANET. It is possible for a malicious node to correctly participate in the route discovery phase but fail to correctly forward data packets. The security solution should ensure that each node indeed forwards packets according to its routing table. This is typically achieved by the reactive approach because attacks on packet forwarding cannot be prevented: an attacker may simply drop all packets passing through it, even though the packets are carefully signed. At the heart of the reactive solutions are a detection technique and a reaction scheme, which are described as follows.

Detection

Normally the wireless channel is open, each node can perform localized detection by overhearing ongoing transmissions and evaluating the behavior of its neighbors. However, its accuracy is limited by a number of factors such as channel error, interference, and mobility. A malicious node may also abuse the security solution and intentionally accuse legitimate nodes. In order to address such issues, the detection results at individual nodes can be integrated and refined in a distributed manner to achieve consensus among a group of nodes. An alternative detection approach relies on explicit acknowledgment from the destination and/or intermediate nodes to the source so that the source can figure out where the packet was dropped.

Localized Detection

Watchdog to monitor packet forwarding on top of source routing protocols like DSR. It assumes symmetric bidirectional connectivity: if A can hear B, B can also hear A. Since the whole path is specified, when node A forwards a packet to the next hop B, it knows B's next hop C. It then overhears the channel for B's transmission to C. If it does not hear the transmission after a timeout, a failure tally associated with B is increased. If the tally exceeds a threshold bandwidth, A sends a report packet to the source notifying B's misbehavior. The same concept but works with distance vector protocols such as ADOV. It adds a next Hopfield in AODV packets so that a node can be aware of the correct next hop of its neighbors. It also considers more types of attacks, such as packet modification, packet duplication, and packet jamming DoS attacks. Each independent detection result is signed and flooded; multiple such results from different nodes can collectively revoke a malicious node of its certificate, thus excluding it from the network.

Ack-Based Detection

The fault detection mechanism is based on explicit acknowledgments. The destination sends back ACKs to the source for each successfully received packet. The source can initiate a fault detection process on a suspicious path that has recently dropped more packets than an acceptable threshold. It performs a binary search between itself and the destination, and sends out data packets piggybacked with a list of intermediate nodes, also called probes, which should send back acknowledgments. The source shares a key with each probe, and the probe list is "onion" encrypted. Upon receiving the packet, each probe sends back an ACK, which is encrypted with the key shared with the source. The source in turn verifies the encrypted ACKs and attributes the fault to the node closest to the destination that sends back an ACK.

3.2 TRUST MODEL IN MANETS

Definition and Properties of Trust

Trust has different meanings in different disciplines from psychology to economy. The definition of trust in MANETs is similar to the explanation in sociology, where trust is interpreted as degrees of the belief that a node in a network (or an agent in a distributed system) will carry out tasks that it should. Due to the specific characteristics of MANETs, trust in MANETs has five basic properties: subjectivity, dynamicity, non-transitivity, asymmetry, and context-dependency.

- **Subjectivity** means that an observer node has a right to determine the trust of an observed node. Different observer nodes may have different trust values of the same observed node.
- **Dynamicity** means that the trust of a node should be changed depending on its behaviors.
- **Non-transitivity** means that if node A trusts node B and node B trusts node C, then node A does not necessarily trust node C.
- **Asymmetry** means that if node A trusts node B, then node B does not necessarily trust node A.
- **Context-dependency** means that trust assessment commonly bases on the behaviors of a node.

Different aspects of actions can be evaluated by different trust. For example, if a node has less power, then it may not be able to forward messages to its neighbors. In this situation, the trust of power in this node will decline, but the trust of security in this node will not be changed due to its state. Reputation is another important concept in trust evaluation. Reputation reflects the public opinions from members in a community. In MANETs, reputation can be a

collection of trust from nodes in the network. Reputation is more global than trust from the perspective of the whole network.

Trust Value Calculation

Based on the definition and properties of trust in MANETs, we evaluate trust in the proposed scheme by a real number, T, with a continuous value between 0 and 1. Although trust and trustworthiness may be different in contexts, in which the trustor needs to consider risk, trust and trustworthiness are treated the same for simplicity in the proposed scheme. In this model, trust value is derived from direct observation method.

Table 3.1: Notation Definition

Notation	Definition
T_{AB}	The total trust value that Node A gives Node B.
T^{S}_{AB}	The trust value that Node A gives Node B based on direct observation of Node A
T^{D}_{AB}	The trust value that Node A gives Node B based on data packets
T^{C}_{AB}	The trust value that Node A gives Node B based on control packets
λ	The weight for the trust value based on direct observation
ρ	The weight for the trust value based on data packets
γ	A factor of punishment which is larger than or equal to 1

In direction observation trust, an observer estimates the trust of his one-hop neighbor based on its own opinion. Therefore, the trust value is the expectation of a subjective probability that a trustor uses to decide whether or not a trustee is reliable. We denote T_S as a trust value from direct observation and can be calculated by Bayesian inference as shown in Equation (3.1). We can get a more realistic and accurate trust value of a node in MANETs.

$$\mathbf{T = \lambda T_S} \qquad (3.1)$$

where λ is a weight assigned to T_S, $0 \leq \lambda \leq 1$.

3.3 PROTOCOL DESIGN

Node Generation and Configuration

The needed number of nodes is generated by using the node command in NS2. The nodes are disseminating in a wireless environment. The random motion is set as true. So, the nodes are moving in a random direction. Each node is considered as an autonomous node. The nodes are configured as to process in MANET environment. The node configuration is done by using node-config command. We have to specify the Channel used by the node, Radio propagation model, Link layer type, Physical layer type, Type of interface queue and the protocol used to route the packets dynamically.

Route Request

Source node will generate a new session key for the association between Source and Destination.

$$\mathbf{S \rightarrow * : [\ RREQ, N_{sq}\ , V_D, V_{SD}, Onion(S)\ ]\ G_{S-}}$$

where RREQ is the packet type identifier; N_{sq} is a sequence number randomly generated by S for this route request; V_D is an encrypted message for the request validation at the destination node; V_{SD} is an encrypted message for the route validation at the intermediate nodes; Onion(S) is a key encrypted onion created by S. The whole RREQ packet is finally signed by S with its group private key G_{S-} .Inter mediate node checks the N_{sq} and the timestamp in order to determine whether the packet has been processed before or not. Then Inter mediate node tries to decrypt the part of V_D with its own private key. In case of decryption failure, Inter mediate node understands that it is not the destination of the RREQ. Inter mediate node will assemble and broadcast another RREQ packet. Destination can decrypt the part of V_D, it understands that it is the destination of the RREQ. Destination can obtain the session key and verify all values.

Route Reply

When Destination receives the RREQ from its neighbor, it will assemble an RREP packet and send it back to neighbor. The format of the RREP packet is defined as follow:

$$\mathbf{D \rightarrow * : (RREP, N_{rt}, \langle K_v , Onion(J)\rangle KJD)}$$

Intermediate nodes are decrypting the reply message if successfully decrypt it identified its valid after it remove the onion layer and send message to next hop. When the RREP packet reaches Source, Source validates the packet in a similar process to the intermediate nodes. If the decrypted onion core NS equals to one of Source issued nonce, Source is the original RREQ source. Then the route discovery process ends successfully. Source is ready to transmit a data along the route indicated.

Trust Evaluation with Direct Observation

Based on the trust model, the framework of the proposed scheme is shown in Figure 3.2. In the trust scheme component, the module of trust evaluation and update can obtain evidence from direct and indirect observation modules and then utilize two approaches, Bayesian inference and DST, to calculate and update the trust values. Next, the trust values are stored in the module of trust repository. Routing schemes in the networking component can establish secure routing paths between sources and destinations based on the trust repository module. The application component can send data through secure routing paths. The trust from direct observation between an observer node *A* and an observed node *B* in this trust scheme can be defined further as in Equation (3.2):

$$\mathbf{T}^{S}_{AB} = \rho \mathbf{T}^{D}_{AB} + (1 - \rho)\mathbf{T}^{C}_{AB} \qquad (3.2)$$

where ρ ($0 \leq \rho \leq 1$) is the weight for data packets; T^{D}_{AB} is the trust value based on data packets; T^{C}_{AB} is the trust value based on control packets.

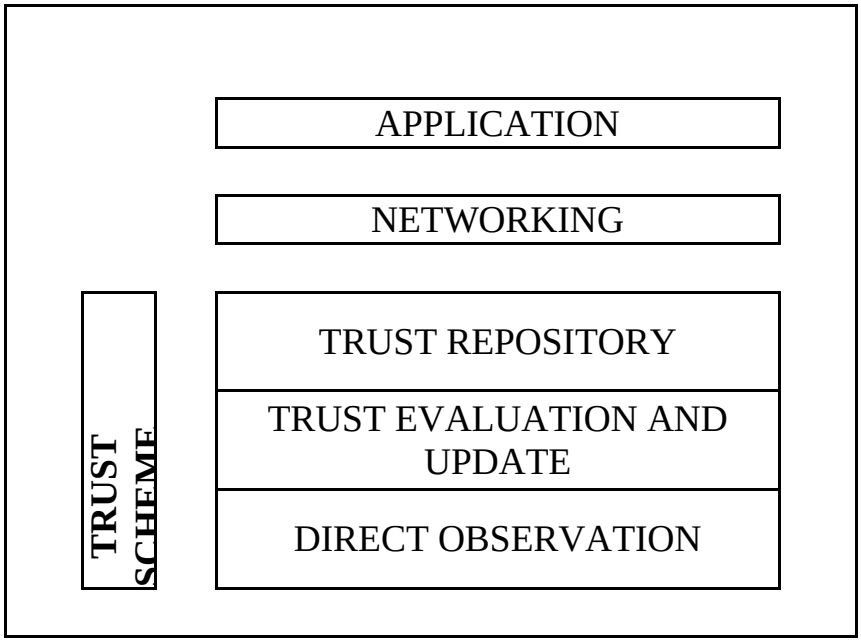

Figure 3.2: Framework of the proposed Scheme

In order to explain the basic procedure of trust evaluation in our scenario, an example network is shown in Figure 3.3. In this example, node 1 is an observer node and node 3 is an observed node. Node 1 sends data messages to node 5 through node 3. When node 3 receives data messages and forwards to node 5, node 1 can overhear it. Then node 1 can calculate the trust value of node 3 based on data messages. The same idea is applied to the control message situation. In the meanwhile, node 1 can collect information from node 2 and node 4, which have interactions with node 3 in order to evaluate the trust value of node 3. This information collected from third party nodes is called indirection observation. In another situation, node 7 sends data messages to node 3, which is the destination node. Node 1 cannot overhear the data messages sent to node 3 in this situation.

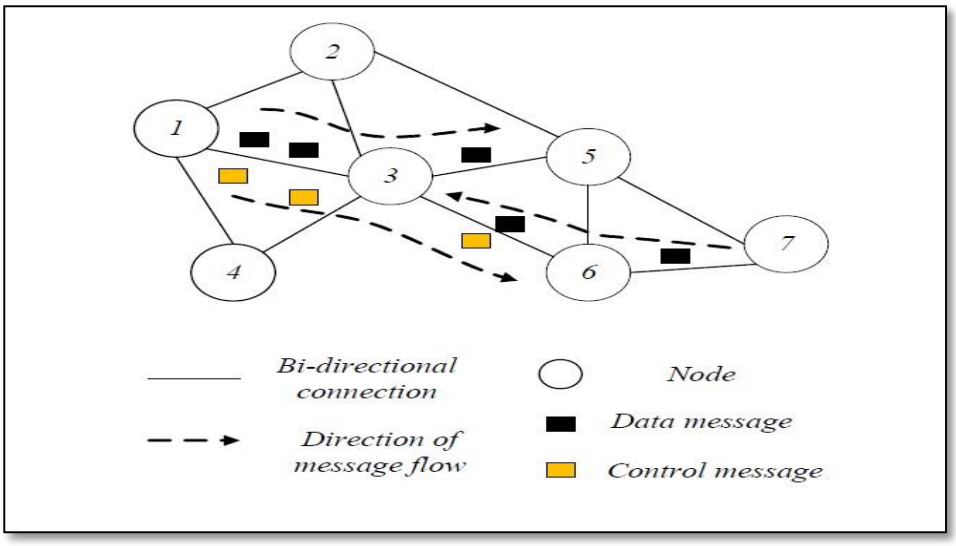

Figure 3.3: Trust Evaluation Scenario

Based on the model presented above, we evaluate trust values with direct observation on two malicious behaviours: dropping packets and modifying packets. In the direct observation, we assume that each observer can overhear packets forwarded by an observed node and compare them with original packets so that the observer can identify the malicious behaviours of the observed node. Therefore, the observer node can calculate trust values of its neighbors by using Bayesian inference, which is a general framework to deduce the estimation of the unknown probability by using observation. As mentioned earlier, the degree of belief is a random variable, denoted by Θ and $0 \leq \theta \leq 1$. From Bayes' theorem, we can derive the following Equation (3.3)

$$f(\theta, y|x) = \frac{p(x|\theta, y)f(\theta,y)}{\int_0^1 p(x|\theta, y)f(\theta,y)} d\theta \qquad (3.3)$$

where x is the number of packets is forwarded correctly; y is the number of packets is received by a node; $p(x|\theta, y)$ is the likelihood function, which follows a binomial distribution as shown in Equation (3.4)

$$p(x|\theta, y) = \binom{y}{x}\theta^x(1-\theta)^{y-x} \qquad (3.4)$$

We assume that the prior distribution, f(θ, y), follows Beta distribution as shown in Equation (3.5),

$$Beta(\theta; \alpha, \beta) = \frac{\theta^{\alpha-1}(1-\theta)^{\beta-1}}{\int_0^1 \theta^{\alpha-1}(1-\theta)^{\beta-1}\, d\theta} \qquad (3.5)$$

where $0 \leq \theta \leq 1$, $\alpha > 0$, $\beta > 0$. Then we can derive this Equation (3.6)

$$f(\theta, y|x) \sim Beta(\alpha + x, \beta + y - x) \qquad (3.6)$$

The expectation of Beta distribution can be formulated as shown in Equation (3.7)

$$E[\Theta] = \frac{\alpha}{\alpha+\beta} \qquad (3.7)$$

Due to reproductivity of f(θ, y|x), the trust value is calculated iteratively. At the beginning, there is no observation. The prior distribution f(θ, y) is Beta(θ; 1, 1) at the beginning. Then we have the Equation (3.8)

$$E_n[\Theta] = \frac{\alpha_n}{\alpha_n+\beta_n} \qquad (3.8)$$

Where

$$\alpha_n = \alpha_{n-1} + x_{n-1}, \beta_n = \beta_{n-1} + y_{n-1} - x_{n-1}, \alpha_0 = \beta_0 = 1, n\, \epsilon Z^|$$

Intuitively, this situation is explained that the trust value of a node is 0.5 at the beginning. That means the node is seemed as neutral when no history records behaviours are established. The value trust can be revised continuously through follow-up observation.

Past experience is also an important factor when trust values are calculated. Recent activities of a node can seriously affect the trust evaluation.

Consider the case where a node has a good history of past experience, but it drops or modifies packets recently. In order to handle this, a windowing scheme is proposed. Using weighted evidence from observation is another method.

In our scheme, we introduce a punishment factor for reputation fading, which focuses on recent activities. The punishment factor is used to give more weights on misbehaviour in the Bayesian framework. Firstly, this can lower the trust of an attacker when it misbehaves. Secondly, the trust of the attack will not recover quickly even if it forwards a large number of packets correctly due to the impact of the punishment factor. This can help the proposed scheme distinguish the malicious node quickly and avoid them disrupting the normal traffic between benign nodes again. The punishment factor is inspired by our daily lives in human society, where a scandal can badly affect a person who has a good reputation. What's more, it is hard to quickly recover a good reputation. The factor of punishment makes the trust evaluation more realistic. The punishment factor, γ, in the formula of trust evaluation in E[Θ] is described as in Equation (3.9):

$$E_n[\Theta] = \frac{\alpha_n}{\alpha_n + \gamma\beta_n} \qquad (3.9)$$

where $\gamma \geq 1$. As the value of γ becomes larger, the trust value declines more. This is because the punishment factor gives more weight to misbehaviour.

Based on this deduction, T^S is defined as in Equation (3.10):

$$T^S = E_n[\Theta] \qquad (3.10)$$

During simulation time the events are traced by using the trace files. The performance of the network is evaluated by executing the trace files. The events are recorded into trace files while executing record procedure. In this procedure,

we trace the events like packet received, Packets lost, Last packet received time etc. These trace values are write into the trace files. This procedure is recursively called for every 0.05 ms. so, trace values recorded for every 0.05 ms.

3.4 PROPOSED BLOCK DIAGRAM

Enhanced authenticated anonymous secure routing is proposed here to overcome the pre mentioned problems. A key-encrypted onion is used to record a discovered route and design an encrypted secret message to verify the RREQ-RREP linkage. Group signature is used to authenticate the RREQ packet per hop, to prevent intermediate nodes from modifying the routing packet. A unified trust management scheme has been integrated with AASR protocol in order to enhance the routing and data security in MANETs. Simulation results have demonstrated the effectiveness of the proposed protocol with improved performance in terms of throughput, packet received ratio, packet loss ratio and delay when compared to the existing ones.

Advantages

- It gives high anonymity protection
- AASR provides higher throughput
- Lower packet loss ratio in different mobile scenarios in the presence of adversary attacks.
- It also provides better support for the secure communications.

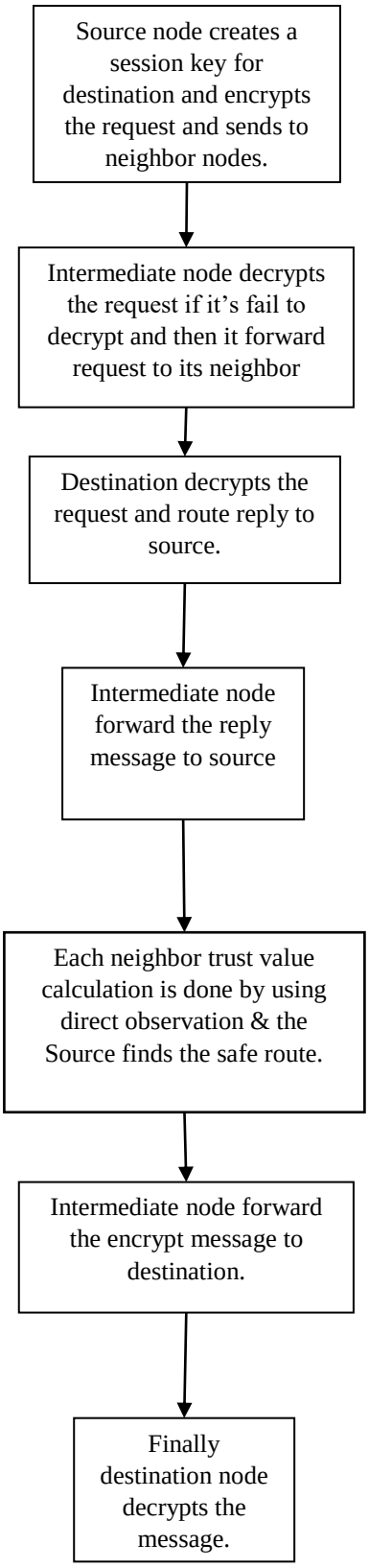

Figure 3.4: Proposed Block diagram

3.5 FLOW DIAGRAM

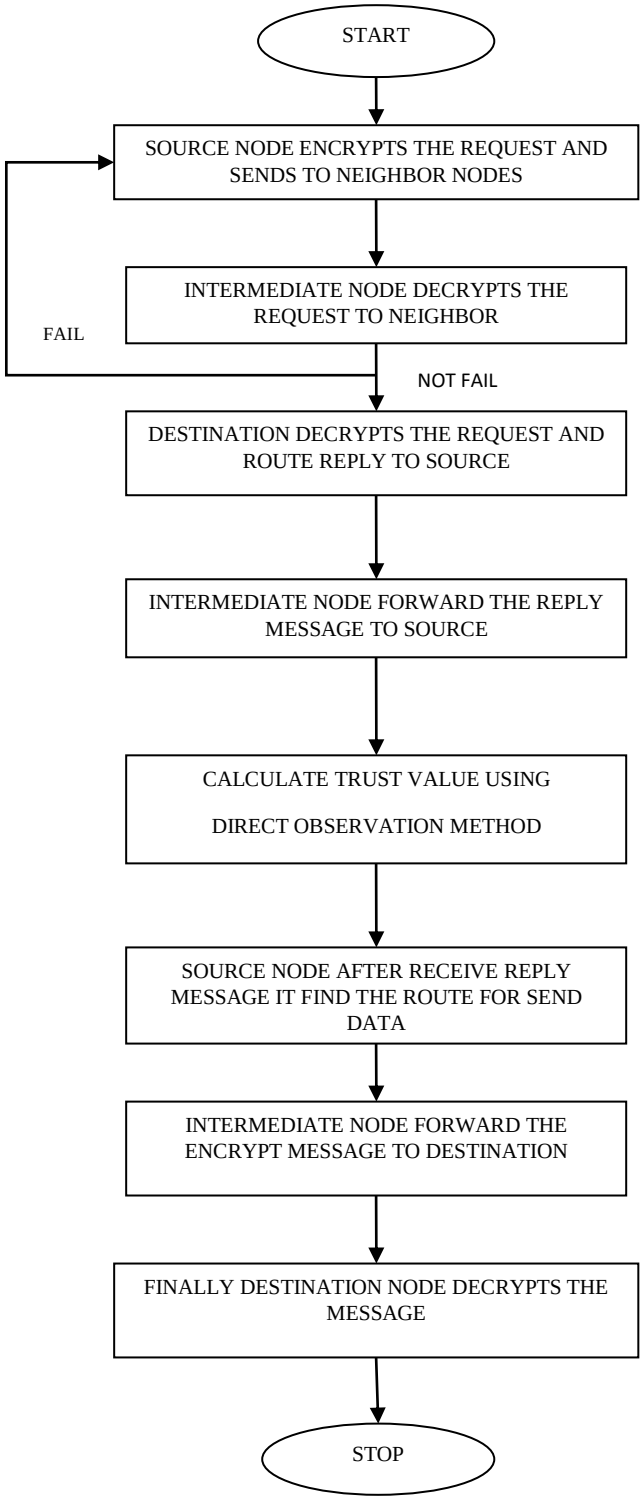

Figure 3.5: Flow Diagram

3.6 ALGORITHM

Step 1: Start the program.

Step 2: Enter the source and destination nodes and enter the session key.

Step 3: The source node encrypts the request message and sends to their neighbor nodes.

Step 4: Intermediate nodes decrypt the request message and sends to the destination nodes.

Step 5: The destination ode decrypt the request message to route reply to source.

Step 6: Intermediate node forward the reply message to source.

Step 7: Source node receives the reply message.

Step 8: Calculate the trust values of packets by using direct observation method.

Step 9: Source node finds the route for sending data based on trust evaluation of nodes.

Step10: Finally destination node decrypts the message.

Step11: Stop the program.

3.7 SIMULATION AND ANALYSIS IN NS2

Introduction to Network Simulator

Network Simulator (NS) is a simulation tool targeted at both wired and wireless (local and satellite) networking research. NS is a very promising tool and is being used by universities and researchers. In this report we provided information how to install NS2 on UNIX and Windows. Then we discussed how to use NS2 to simulate wired and wireless networks. A simple but limited method is to combine the existing components with OTcl scripts; a complex but

powerful method is to implement new components into NS2 using C++. We discussed both of the two methods. Finally, the methods to animate (using NAM) and to analyze (using Xgraph/GNUplot) the simulation results are presented.

Network Simulator (NS), a discrete event simulator targeted at networking research is widely used in universities and companies by researchers. NS provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. The latest version of NS is 2.26 (NS2). It implements network protocols such as FTP and Telnet, routing algorithms such as SPF and DV, and 'lower' layers such as logic link (LL) and media access control (MAC). NS began as a variant of the REAL network simulator in 1989. In 1995 NS development was supported by DARPA through the VINT project at LBL, Xerox PARC, UCB, and USC/ISI. Currently NS development is support through DARPA with SAMAN and through NSF with CONSER. NS has always included substantial contributions from other researchers, including ACIRI, UCB, CMU, and Sun Microsystems.

Nam began at LBL. The Nam development effort was an ongoing collaboration with the VINT project. Currently, it is being developed at ISI as part of the SAMAN and Conser projects. The simple way NS2 can be used is for studying the property of a well-known protocol. In this case, a script language OTcl is used to glue the network components (nodes, links, agents, applications, etc) provided by NS2, configure the parameters (band-width, delay, routing protocol, etc) and launch activities (data transfer, topology change, etc). NS2 will read these configurations; simulate each network event, and record events and statistics in to trace files. After the simulation, Nam can demonstrate the events in a visualized way. For the simple usage of NS2, an understanding to the simulation framework is necessary.

When NS2 is used to simulate a new protocol, e.g., an ad hoc wireless routing protocol, the simple way is not enough. The advanced way is to hack the source of NS2 with C++, define new network components, rebuild the whole system and run the customized version of NS2. For the advanced usage of NS2, knowledge of the simulator implementation is required.

Simulation Framework

NS2 simulation is about the processing of packets. Packets are transported among network nodes via network links. These packets are generated by network applications (data packet) or network protocols (control packet). An event scheduler is in charge of ordering all the packets (or events) by their arriving time and let nodes, links and agents to handle these packets in sequence. A full trace of each packet or their statistics information can be stored in a trace file, and used to generate graphs or animations. In this section, the concept of node, link, agent, and scheduler will be introduced, as well as their usage in an OTcl script to configure a specific NS2 simulation.

The following script segment creates two nodes. When a node is created, it is automatically assigned an address and a default routing module. Nodes are the sources and destinations of the packets. In the script, $ns is the scheduler object and node is its method to create a node, $n0 and $nl hold the reference to the newly created node object.

Set n0 [$ns node]

Set ni [$ns node]

The following script segment creates one link to connect these two nodes with lM bandwidth and l0ms delay. A link receives packets from one end, compute the delay and transmission time, and send them to the other end after the time elapsed. If two or more packets are going through the same link, the later packet should wait in a queue until all previous packets are sent. If the

number of packets waiting exceeds the queue limit, new packets sent to this link are dropped.

$ns duplex-link $n0 $n1 1Mb 10ms Drop Tail

The default routing algorithm is centralized link state algorithm (like Dijkstra's SPF). The route from one node to each other node is computed by the simulator and stored in the routing table of this node. From user's point of view, each node just 'knows' how to reach another node in the network.

Agent: Network Protocol and Applications

When the network topology is created, agents can be attached to nodes to generate packets and put them onto the network. There are agents in different network protocol layers. In the transportation layer, there are TCP and UDP agents. The following OTcl script segment creates a UDP agent and put it on node $n0. When an agent is attached to a node, it sends and receive packet through this node.

Set udp0 [new Agent/UDP]

$ns attach-agent $n0 $udp0

In the application layer, there are agents for well known protocols like FTP, Telnet, HTTP, etc; there are also special agents for simulation use, like CBR, which generate continues data streams. The following OTcl script-segment creates a CBR agent and attaches it to the UDP agent. When an agent is attached to another agent, it sends and receives data through this agent.

Set cbr0 [new Application/Traffic/CBR]

$cbr0 attach-agent $udp0

Agents are usually created in pairs, put in different nodes and exchange data between each other. The following OTcl script segment creates a Null

agent on node $n1 and connect it to the UDP agent on node $n0. After the agents are connected, the data generated by the CBR agent and passed to the UDP agent will go though node $n0, the link between node $n0 and $n1, node $n1, and received by the Null agent. The Null agent is like a null device, it discards any packet it receives.

Set null0 [new Agent/Null]

$ns attach-agent $n1 $null0

$ns connect $udp0 $null0

There are also agents in other layers, e.g., the router agents when dynamic routing is used. However, these agents are always hidden in the inner structure of nodes, and created automatically by the simulator.

Scheduler

If nodes and links is the scene in the stage, agents are the actors; then the scheduler is the director. The scene and actors don't know how to perform unless the director tells them. The scheduler is a build-in object in NS2 user can tell it when to do what. The following OTcl script segment asks the scheduler to open the traffic generator at 0.5 second of the simulation and close it at 4.5 second. The script segment also told the scheduler the whole simulation will last for 5.0 seconds.

$ns at 0.5 "$cbr0 start" $ns at 4.5 "$cbr0 stop"

$ns at 5.0 "finish"

Actually, a scheduler's work is much more than that. Every time a packet is received by an agent or a link, a delay time representing the processing time or network delay is computed and actually the packet is handed to the scheduler as an event. The scheduler orders all the events by time and fires them one by one; i.e., when the time comes, the agent or link will send the early received

packet to its destination. There are also other events than packets, e.g., a TOP agent may set a timer for its time window and re-send a packet when timeout.

$ns run

The above OTcl script segment will start the simulation; i.e., the scheduler begins to fire the events one by one, according to their time stamps. In NS2, the scheduler is non-real-time by default. The time stamps are not physical time in the simulation, but used to order the events. So a 5.0-second simulation may take 2.0 seconds or 1 hour, de-pending on the complexity of the topology and the number of events fired during the simulation.

CHAPTER 4
RESULTS AND DISCUSSION

4.1 NODE CONFIGRATION

This Figure 4.1 shows several nodes. Each node has some transmitting range. The route is identified by using the trust based routing protocol. The simulation begins with the generation and configuration of the needed number of nodes. The Figure 4.1 below shows a MANET model of 30 nodes.

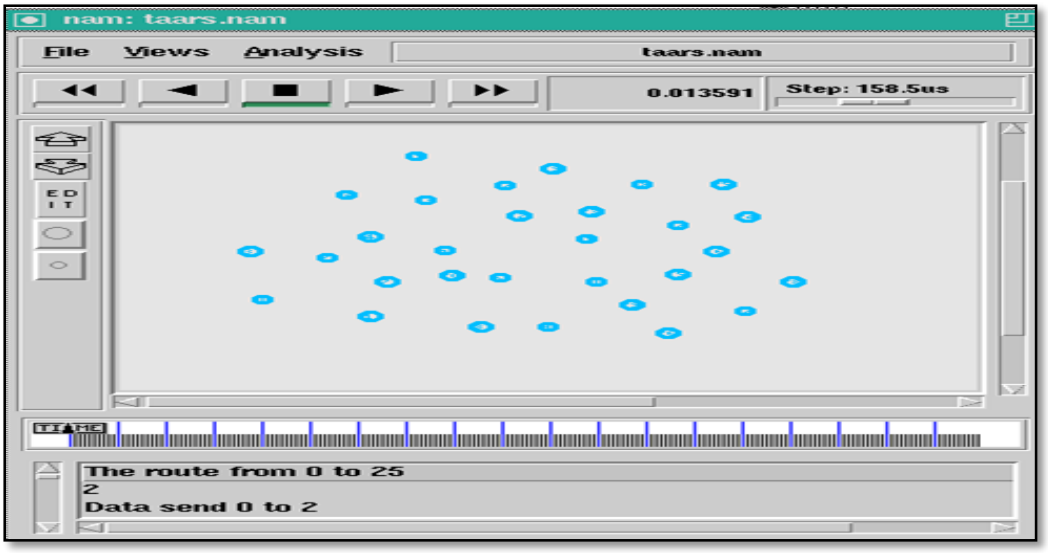

Figure 4.1: Node configuration

4.2 ROUTE REQUEST AND ROUTE REPLY

The route discovery phase comprises of RREQ (Route request message) sent by the source node to destination node and RREP (Route reply message) from destination node back to source node via neighbor and intermediate nodes which is shown in Figure 4.2.

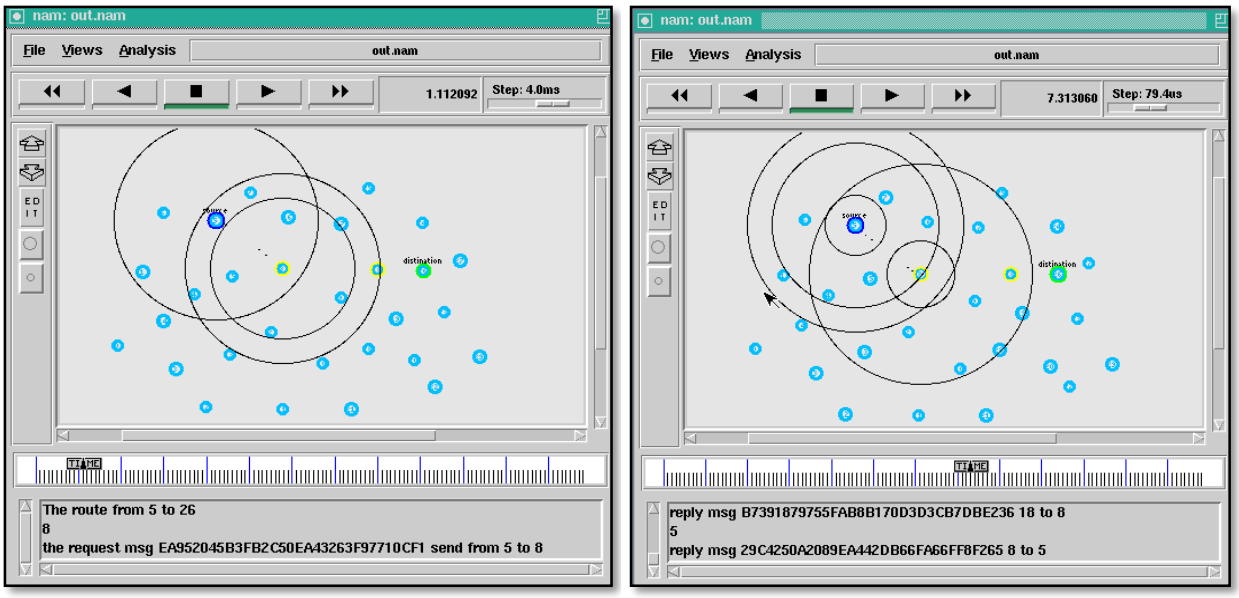

Figure 4.2: Route Request and Reply

4.3 TRUST GET PHASE AND DATA TRANSMISSION

Once the secure route is identified the source node encrypts the data and transmits it via established route based the trust values calculated for each intermediate node and the same is decrypted at the destination. Figure 4.3 shows the encryption at source node; Figure 4.4 shows the encryption and forwarding of data by intermediate node and the Figure 4.5 shows the decryption at destination node.

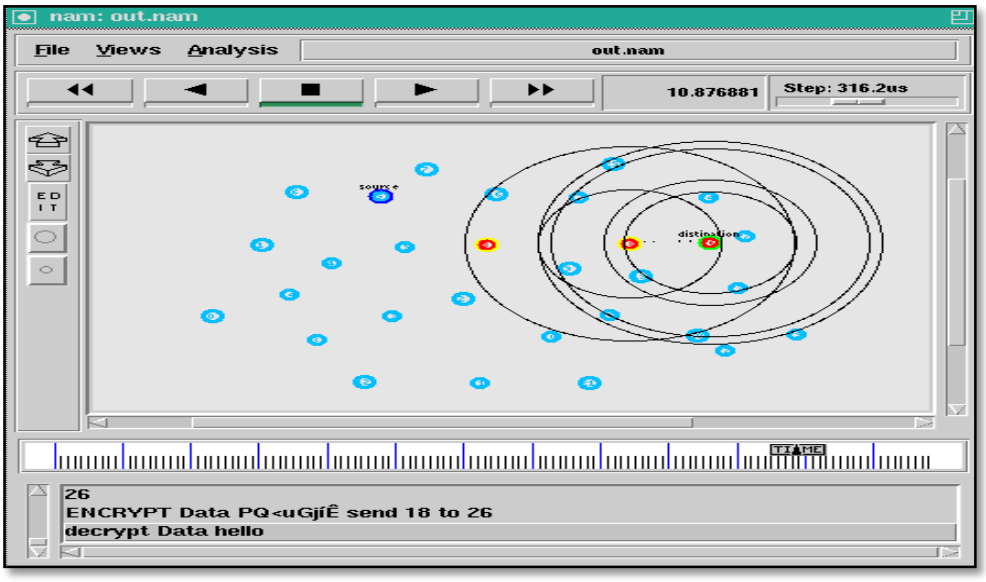

Figure 4.3: Encryption at Source node

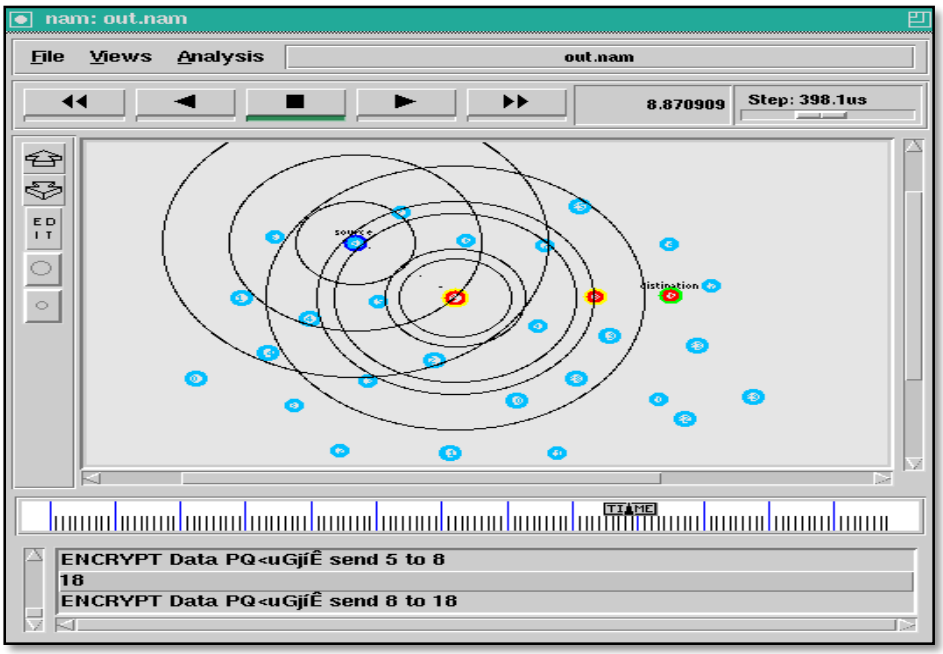

Figure 4.4: Encryption and forwarding of data by Intermediate node

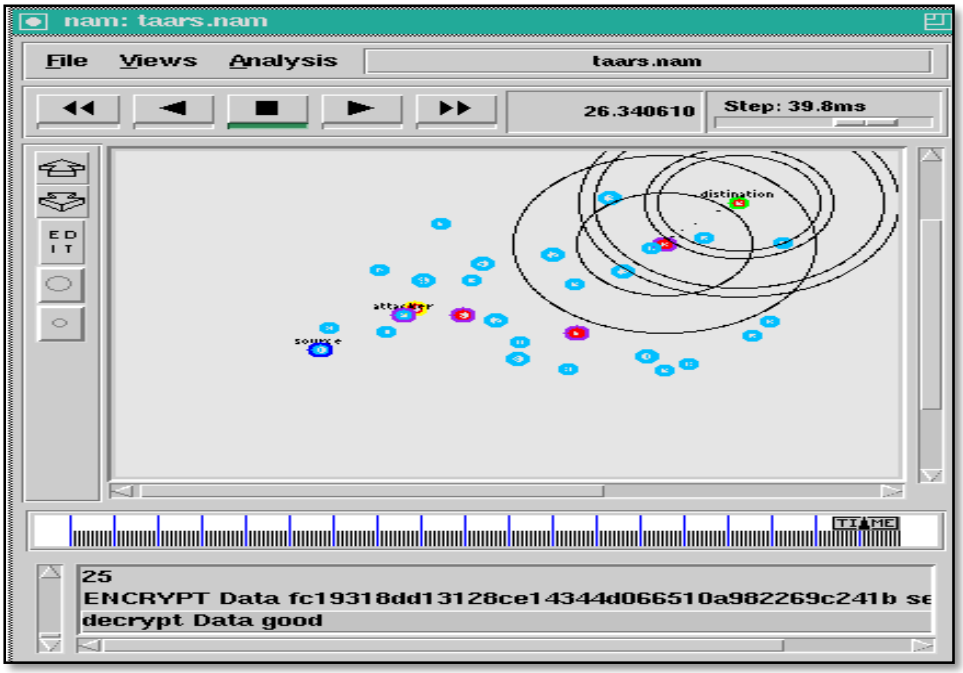

Figure 4.5: Decryption at Destination node

4.4 SIMULATION RESULTS

The proposed system is evaluated and compared with the existing systems based on the factors such as topology and traffic, attack models and the end results.

- Topology and Traffic – Network area is 1501m × 600m with 30 nodes initially and uniformly distributed. The radio uses the two ray ground reflection propagation model.
- Attack model – It is assumed that only the intermediate nodes along a route may become malicious. A malicious node will randomly drop routing packets. ANODR and AODV will suffer more packets losses than the proposed system. The attacks are simulated in the following way: AODV takes no action against the attack; ANODR acts in its routing maintenance procedures. The proposed system can detect the malicious node via the group signature, and get rid of the attackers in the routing tables.
- When the number of malicious nodes increases, the average throughput of protocols such as AODV, ANODR decreases obviously. Since the proposed system has the ability to detect the packet dropping attack, it outperforms ANODR and AODV.

Throughput

Throughput may be defined as the rate of successful message delivery in a given period of time in a communication network. The Figure 4.6 shows that the throughput results obtained for the protocol integrated with a trust scheme is more when compared to the existing protocol.

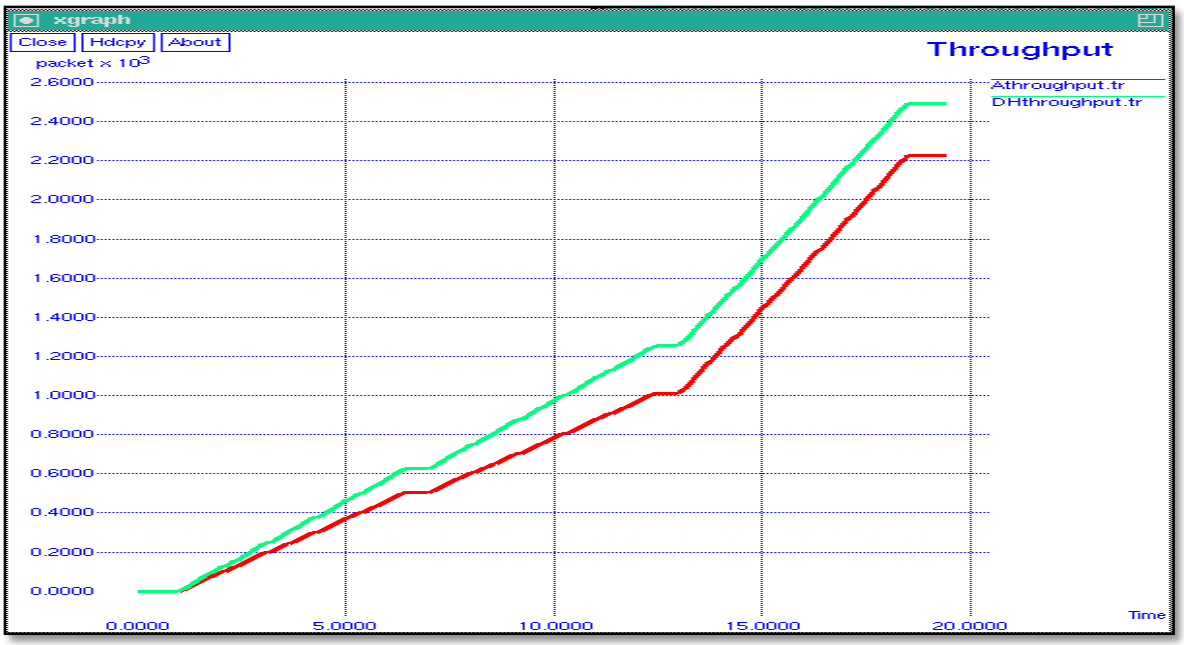

Figure 4.6: Throughput

Packet Received Ratio

Packet received ratio is defined as the ratio of the number of received data packets to the number of sent data packets. The greater value of the packet received ratio means the better performance of the protocol which is illustrated in the Figure 4.7 below.

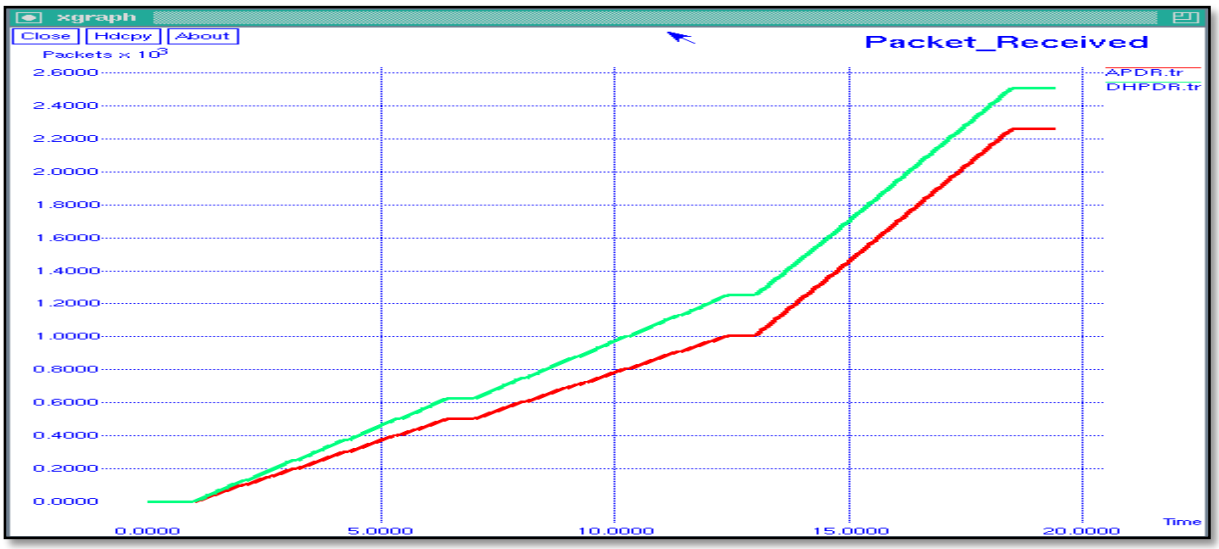

Figure 4.7: Packet received ratio

Packet Loss Ratio

Packet loss refers to the total number of packets lost during simulation. It may be given as,

Packet lost = Number of packets sent – Number of packets received

The Figure 4.8 below shows that the packet loss ratio of the proposed protocol is less than the existing one which indicates the better performance of the proposed protocol.

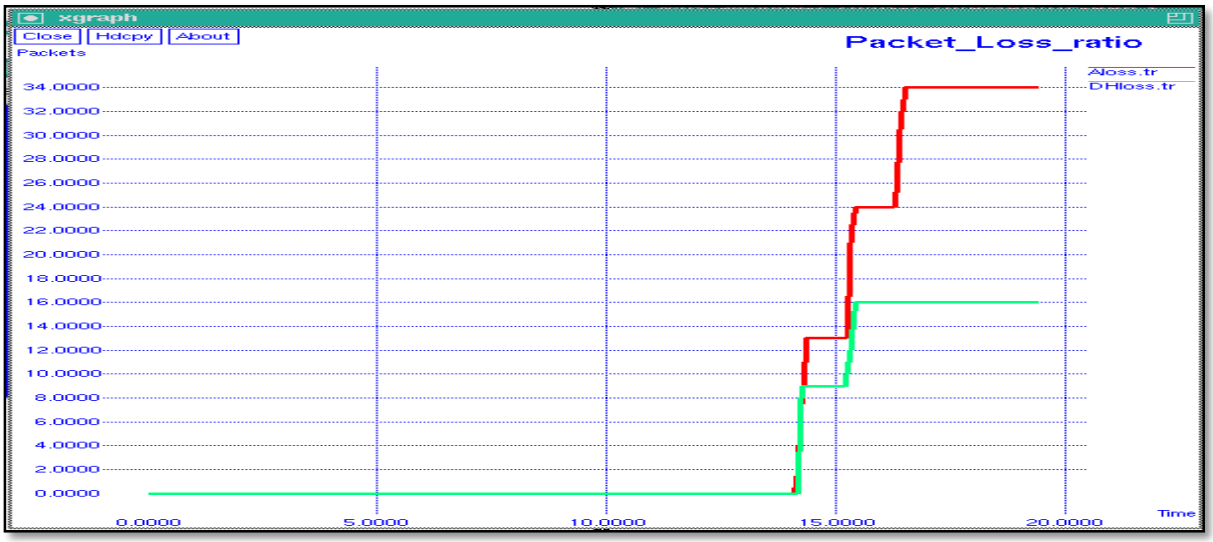

Figure 4.8: Packet loss ratio

Delay

End to end delay refers to the average time taken by the data packet to arrive in the destination. It also includes the delay caused by the route discovery process and the queue in data packet transmission. Only the data packets that are successfully delivered are counted. It is given by

$\sum$ (arrival time –send time) / $\sum$ Number of connections

The Figure 4.9 below shows lower value of end to end delay which proves the better performance of the proposed protocol.

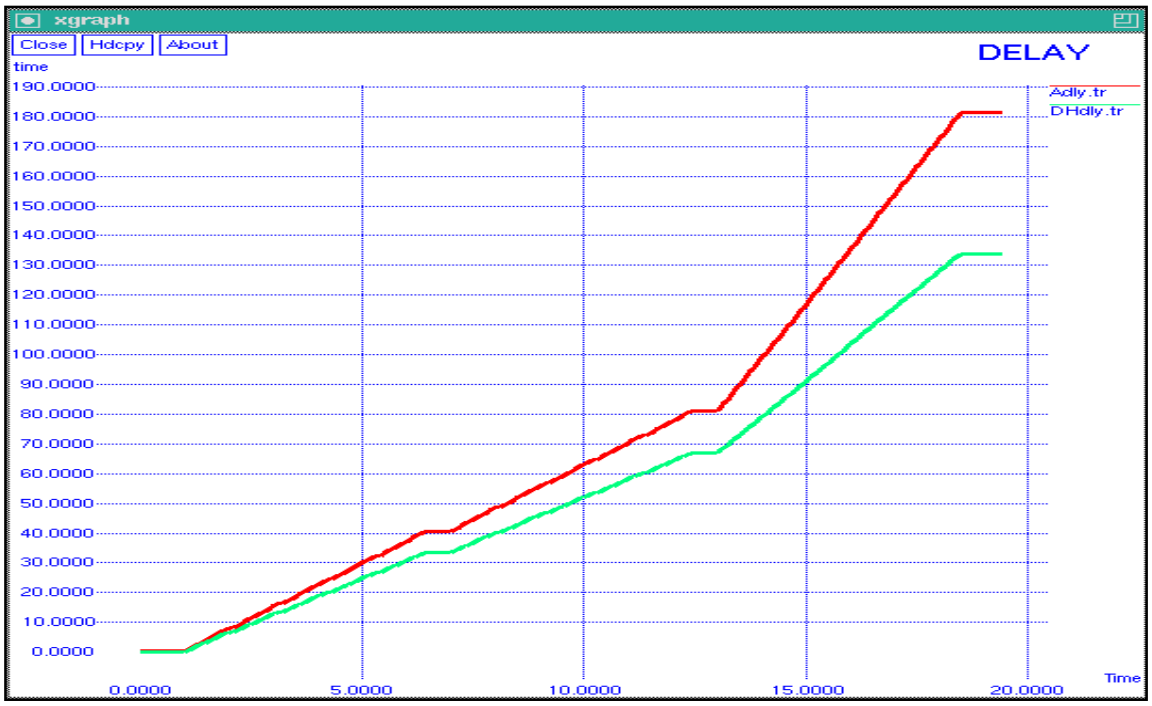

Figure 4.9: Delay

Trust Value of Nodes

This graph in Figure 4.10 shows the trust values of the nodes in a trust based routing protocol. The graph is the plot of number of nodes and the trust values.The trust from direct observation between an observer node *A* and an observed node *B* in this trust scheme can be defined further as:

$$T^{S}_{AB} = \rho T^{D}_{AB} + (1 - \rho) T^{C}_{AB}$$

where ρ $(0 \leq \rho \leq 1)$ is the weight for data packets; T^{D}_{AB} is the trust value based on data packets; T^{C}_{AB} is the trust value based on control packets.

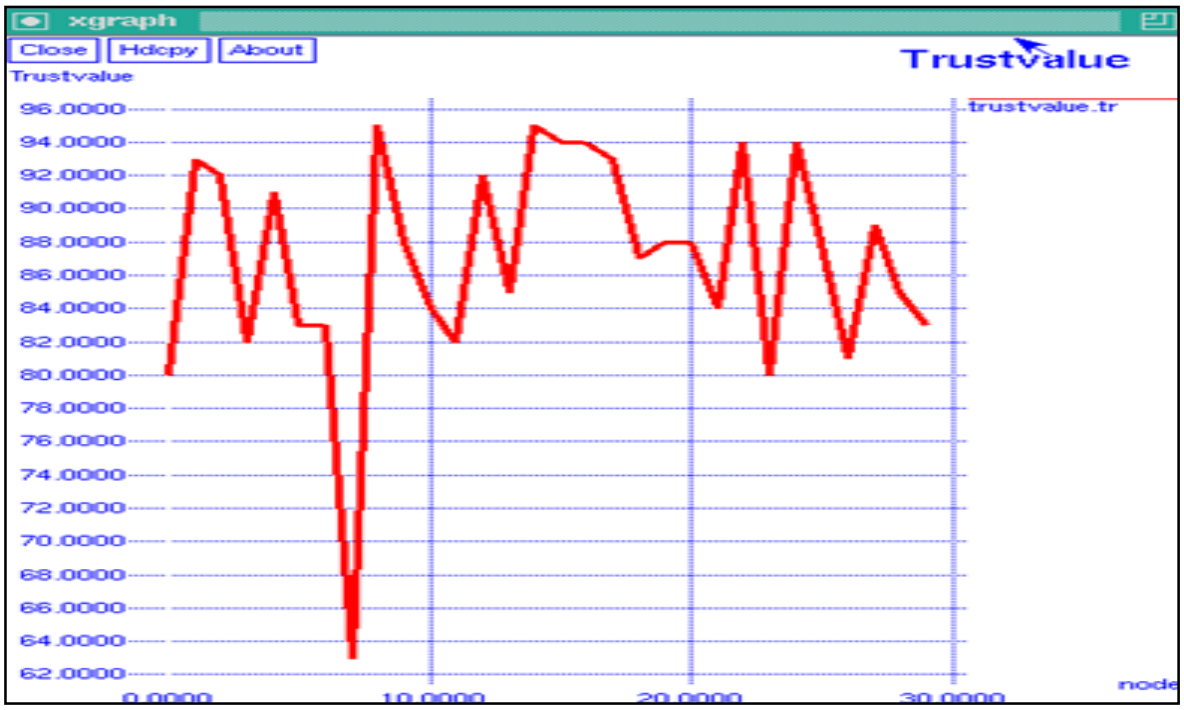

Figure 4.10 Trust value of nodes

The simulation results shows the effectiveness of the proposed protocol with improved performance when compared to the existing protocols in terms of throughput, packet received ratio, packet lost ratio and time delay.

CHAPTER 5
CONCLUSION AND FUTURE WORK

In this work, a unified trust management scheme integrated with AASR protocol that enhances the security of MANETs was proposed. Using recent advances in uncertain reasoning and through Bayesian inference method, an evaluation of the trust values of observed nodes in MANETs is performed. Misbehaviors such as dropping or modifying packets can be detected in this scheme through trust values by direct observation. Nodes with low trust values will be excluded by the AASR routing algorithm. Therefore, secure routing path can be established in malicious environments. Based on the proposed scheme, more accurate trust can be obtained by considering different types of packets and other important factors such as buffers of queues and states of wireless connections, which may cause dropping packets in friendly nodes. The simulation results of MANET routing scenario positively support the effectiveness and performance of the proposed scheme, which improves throughput and packet delivery ratio considerably.

In future work we can use advanced encryption algorithms to achieve higher security for the in-transit data and extending the proposed scheme to MANETs with cognitive radios. Indirect observation method (Dempster Shafer Theory) of calculating the trust value of nodes can be combined with direct method to get even more accurate values.

APPENDIX

A.1: main.tcl

```
# explicitly setup our main window

wm geometry  . 850x350+300+200
wm title  .   "AASR-DH "
# setup the frame stuff

destroy .myArea
set f [frame .myArea -borderwidth 5 -background blue]
pack $f -side top -expand true -fill both

# create a menubar

destroy .menubar
menu .menubar
. config -menu .menubar

#  create a pull down menu with a label
set File2 [menu .menubar.mFile2]
.menubar add cascade -label "AASR "  -menu  .menubar.mFile2
set File3 [menu .menubar.mFile3]
.menubar add cascade -label "PerformanceEvaluation"  -menu  .menubar.mFile3

set close [menu .menubar.sFile]
.menubar add cascade -label Quit  -menu  .menubar.sFile
```

```
# add the menu item


$File2 add command -label Run_AASR-Data -command {exec ./ns aars.tcl &}

$File2 add command -label Run_Simulation -command {exec nam aars.nam &}

$File2 add command -label Run_TAASR-Data -command {exec ./ns taars.tcl
&}

$File2 add command -label Run_Simulation -command {exec nam taars.nam
&}

$File3 add command -label Packet_Received -command {exec xgraph APDR.tr
tAPDR.tr    -x "Time" -y "Packets"  -bg "white" -fg "blue" -t Packet_Received -
lw 3 &}

$File3 add command -label Packet_Loss_ratio -command {exec xgraph Aloss.tr
tAloss.tr -x "Time" -y "Packets" -bg "white" -fg "blue" -t Packet_Loss_ratio -lw
3 &}

$File3 add command -label Throughput -command {exec xgraph
Athroughput.tr tAthroughput.tr -x "Time" -y "packet" -bg "white" -t Throughput
-fg "blue" -lw 3 &}

$File3 add command -label DELAY -command {exec xgraph Adly.tr tAdly.tr -
x "Time" -y "packet" -bg "white" -t DELAY -fg "blue" -lw 3 &}

$File3 add command -label Trustvalue -command {exec xgraph  trustvalue.tr -x
"node" -y "Trustvalue" -bg "white" -t Trustvalue -fg "blue" -lw 3 &}

$close add command -label Quit -command exit
```

A.2: Simulation Parameters Setup

```
set val(chan)   Channel/WirelessChannel    ;# channel type

set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model

set val(netif)  Phy/WirelessPhy            ;# network interface type

set val(mac)    Mac/802_11                 ;# MAC type

set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type

set val(ll)     LL                         ;# link layer type

set val(ant)    Antenna/OmniAntenna        ;# antenna model
```

```
set val(ifqlen) 50                ;# max packet in ifq
set val(nn)     30                ;# number of mobilenodes
set val(rp)     DSDV              ;# routing protocol
set val(x)      1501              ;# X dimension of topography
set val(y)      600               ;# Y dimension of topography
set val(stop)   30.0              ;# time of simulation end
```

A.3: Mobile Node Parameter Setup

```
$ns node-config -adhocRouting  $val(rp) \
                -llType        $val(ll) \
                -macType       $val(mac) \
                -ifqType       $val(ifq) \
                -ifqLen        $val(ifqlen) \
                -antType       $val(ant) \
                -propType      $val(prop) \
                -phyType       $val(netif) \
                -channel       $chan \
                -topoInstance  $topo \
                -agentTrace    ON \
                -routerTrace   ON \
                -macTrace      ON \
                -movementTrace ON
proc Trust { } {
        global ns array names n sor dest route null4 route2
set sn $sor
set dn $dest
set s $sn
        set sink $dn
        set r 0
```

```
set t [$ns now]
$ns at [$ns now] "$n($s) add-mark mo blue"
$ns at [$ns now] "$n($sink) add-mark mo green"
$ns at [$ns now] "$n($s) label source"
$ns at [$ns now] "$n($sink) label SINK"
$ns at $t "$ns trace-annotate \" The route from $s to $sink\""
while {$s!=$sink} {
        set ni [lindex $route2($sn,$dn) 0]
if {$r==0} {
set in $ni
}   else {
        set in [lindex $route($sn,$dn) $r]
        }
        puts "in:$in"
        $ns at $t "$ns trace-annotate \" $in \""
        $ns attach-agent $n($in) $null4
        # transPower $in
        set cbr01 [attach-cbr-traffic $n($s) $null4 150 0.006]
        $ns at $t "$cbr01 start"
        $ns at $t "$ns trace-annotate \" Data send $s to $in \""
        $ns at [$ns now] "$n($in) add-mark mo purple"
        $ns at [expr $t+1.0] "$cbr01 stop"
        set t [expr $t+1.5]
        set s $in
        incr r
        puts "r:$r"
                }
        puts "Time:$t"
        # $ns at [expr $t+0.3] "Transmission1"
```

```
            $ns at [expr $t + 0.5] "RREQ"
            $ns at [expr $t + 0.5] "$n($ni) label attacker"
        }
```

A.4: Distance Calculation

```
    set nn 30
     for {set i 0} {$i<$nn} {incr i} {
            set neighborlist($i) [list]
            set x_pos1 [$n($i) set X_]
            set y_pos1 [$n($i) set Y_]
            puts "Distance from node $i"
            puts "****************************************"
            puts $dis "***********************************"
            puts "FROM          TO     DISTANCE"
            puts $dis "FROM   TO     DISTANCE"
            puts "****************************************"
            puts $dis "***********************************"

            for {set j 0} {$j<$nn} {incr j} {
                  if {$j!=$i} {
                  set x_pos2 [$n($j) set X_]
                        set y_pos2 [$n($j) set Y_]
                  set x_pos [expr $x_pos1-$x_pos2]
                  set y_pos [expr $y_pos1-$y_pos2]
                  set v [expr $x_pos*$x_pos + $y_pos*$y_pos]
                  set d [expr sqrt($v)]
                  set nd($i,$j) $d
                  puts $dis "Node$i  Node$j        $d"
                  puts "Distance from $i to $j:$d"
                  if {$d<220} {
                        $n($i) add-neighbor $n($j)
                  }}}
```

A.5: Trust Detection Procedure

```
proc Trust { } {
        global ns array names n sor dest route null4 route2
set sn $sor
set dn $dest
set s $sn
        set sink $dn
        set r 0
        set t [$ns now]
        $ns at [$ns now] "$n($s) add-mark mo blue"
        $ns at [$ns now] "$n($sink) add-mark mo green"
        $ns at [$ns now] "$n($s) label source"
        $ns at [$ns now] "$n($sink) label SINK"
        $ns at $t "$ns trace-annotate \" The route from $s to $sink\""
        while {$s!=$sink} {
                set ni [lindex $route2($sn,$dn) 0]
        if {$r==0} {
        set in $ni
        }   else {
                set in [lindex $route($sn,$dn) $r]
                }
                puts "in:$in"
                $ns at $t "$ns trace-annotate \" $in \""
                $ns attach-agent $n($in) $null4
                # transPower $in
                set cbr01 [attach-cbr-traffic $n($s) $null4 150 0.006]
                $ns at $t "$cbr01 start"
                $ns at $t "$ns trace-annotate \" Data send $s to $in \""
                $ns at [$ns now] "$n($in) add-mark mo purple"
```

```
        $ns at [expr $t+1.0] "$cbr01 stop"
        set t [expr $t+1.5]
        set s $in
        incr r
        puts "r:$r"
                }
        puts "Time:$t"
        # $ns at [expr $t+0.3] "Transmission1"
        $ns at [expr $t + 0.5] "RREQ"
        $ns at [expr $t + 0.5] "$n($ni) label attacker"
}
```

REFERENCES

1. D. Boneh, X. Boyen, and H. Shacham, Aug. 2004. “Short group signatures,” in Proc. Int. Cryptology Conf. (CRYPTO’04)

2. S. Corson and J. Macker, Jan. 1999, “Mobile ad hoc networking (MANET): routing protocol performance Issues and evaluation considerations,” IETF RFC2501.

3. K. E. Defrawy and G. Tsudik, Dec. 2011, “Privacy-Preserving Location-Based On-Demand Routing in MANETs,” IEEE Journal on Selected Areas in Communications, vol. 29, no. 10, pp. 1926–1934.

4. J. Kong and X. Hong, Jun. 2003, “ANODR: ANonymous On Demand Routing with Untraceable Routes for Mobile Ad hoc networks,” in Proc. ACM MobiHoc’03, pp. 291–302.

5. K.K.Lakshmi Narayanan and Fidal Castro, March 2012, “High Security for Manet Using Authentication and Intrusion Detection with Data Fusion”, International Journal of Scientific & Engineering Research Volume 3, Issue 3.

6. C. Perkins, E. Belding-Royer, S. Das, 2003, “RFC 3561 - Ad hoc On-Demand Distance Vector (AODV) Routing,” Internet RFCs.

7. H. Shen and L. Zhao, 2013,“ALERT: An Anonymous Location-based Efficient Routing Protocol in MANETs,” IEEE Trans. on Mobile Computing, vol. 12, no. 6, pp. 1079–1093.

8. R. Song, L. Korba, and G. Yee, Nov. 2005, “AnonDSR: efficient anonymous dynamic source routing for mobile ad hoc networks,” in Proc. ACM Workshop Security of Ad Hoc and Sensor Networks (SASN’05).

9. Z. Wan, K. Ren, and M. Gu, May 2012, “USOR: An Unobservable Secure On-Demand Routing Protocol for Mobile Ad Hoc Networks,” IEEE Trans. on Wireless Communication, vol. 11, no. 5, pp. 1922–1932.

10. M. Yu and K. Leung, Apr. 2009, “A Trustworthiness-based QoS routing protocol for ad hoc networks,” IEEE Trans. on Wireless Comms., vol. 8, no. 4, pp.1888–1898.

11. F. R. Yu, H. Tang, P. Mason, and F. Wang, Dec. 2010, “A hierarchical identity based key management scheme in tactical mobile ad hoc networks,” IEEE Trans. on Network and Service Management, vol. 7, pp. 258 – 267.

12. M. Yu, M. C. Zhou, and W. Su, Jan. 2009, “A secure routing protocol against Byzantine attacks for MANETs in adversarial environment,” IEEE Trans. on Vehicular Tech., vol. 58, no. 1, pp. 449–460.

13. Y. Zhang, W. Liu, W. Lou, and Y. G. Fang, , Sept. 2006, “MASK: Anonymous On-Demand Routing in Mobile Ad hoc Networks,” IEEE Trans. on Wireless Comms., vol. 5, no. 9, pp. 2376–2386.

14. Zhexiong Wei, Helen Tang, F. Richard Yu, Maoyu Wang, and Peter Mason,2014,” Security Enhancements for Mobile Ad Hoc Networks with Trust Management Using Uncertain Reasoning”, IEEE Transactions on Vehicular Technology.