Macrothink Institute™

Network Protocols and Algorithms
ISSN 1943-3581
2011, Vol. 3, No. 4

Impact of Security Attacks on a New Security Protocol for Mobile Ad Hoc Networks

K. Sahadevaiah,

Department of Computer Science & Engineering, JNTUK College of Engineering,

Jawaharlal Nehru Technological University, KAKINADA-533003, AP (India)

Tel: +91 9502661429, E-mail: ksd1868@gmail.com

Prasad Reddy P.V.G.D.,

Department of Computer Science and Systems Engineering, AU College of Engineering,

Andhra University, VISAKHAPATNAM-5300003, AP (India)

Tel: +91 9393112979, E-mail: prof.prasadreddy@gmail.com

Received: December 10, 2011 Accepted: December 28, 2011 Published: December 31, 2011

DOI: 10.5296/npa.v3i4.1364 URL: http://dx.doi.org/10.5296/npa.v3i4.1364

Abstract

A mobile ad hoc network (MANET) is a self-organized wireless short-lived network consisting of mobile nodes. The mobile nodes communicate with one another by wireless radio links without the use of any pre-established fixed communication network infrastructure. The mobile nodes are vulnerable to different types of security attacks that allow interception, injection, and interference of communication among nodes. Possible damages include leaking secret information, message contamination and node impersonation. MANETs need secure routing protocols to prevent possible security attacks. In this paper, we evaluate the performance of a new security protocol against various known and unknown malicious node attacks. Simulation results have shown that the proposed security protocol resists against malicious nodes with low implementation complexity.

Keywords: mobile ad hoc networks, self-organization, cryptography, security attacks, key management, security protocol.

1. Introduction

A *Mobile Ad hoc NETwork* (MANET) is a self-organized wireless short-lived network consisting of mobile nodes. The mobile nodes communicate with one another by wireless radio links without the use of any pre-established fixed communication network infrastructure. Typical MANET nodes are Laptops, PDAs, Pocket PCs, Cellular Phones, Internet Mobile Phones, and Palmtops. These devices are typically lightweight and battery operated [1][2][3].

The mobile nodes are vulnerable to different types of security attacks than conventional wired and wireless networks. This is due to their open medium, dynamic network topology, absence of central administration, distributed cooperation, constrained capability, and lack of clear line of defense. The unconstrained nature of a wireless medium of MANETs allows the attackers for interception, injection, and interference of communication among nodes. Possible damages include leaking secret information, message contamination and node impersonation [4].

To prevent possible security attacks, MANETs need secure routing protocols. There exist various secure routing protocols, such as SAR, ARAN, SAODV, SRP, ARIADNE, SEAD, SMT, SLSP, CONFIDANT, etc. in the literature and widely evaluated for efficient routing of packets [3][4]. But these protocols are either too expensive or have unrealistic requirements. They consume a lot of resources, and delay or even prevent successful exchanges of routing information. Security extensions for existing routing protocols do not contain important performance optimizations. Inclusion of optimistic approaches provides a better trade-off between security and performance. Resource limitations of mobile devices, such as memory, computation, communication and energy, need to be carefully considered in the solution [5][6][7].

The major aim of this paper is to evaluate the performance of a new security protocol against various known and unknown security attacks. The proposed security protocol solutions rely on private-public key cryptography and digital signatures to achieve the security goals like message integrity, data confidentiality, and end-to-end authentication. In the proposed scheme, the nodes are not responsible for issuing other nodes’ certificates. Every intermediate node checks the neighbor's digital signatures, which guarantee that no single node modifies the public key certificate information during the distribution process. The reason is that the certificates are distributed securely to the neighboring nodes with the symmetric key encryption.

The rest of the paper is organized as follows: Section 2 reviews security attacks. Section 3 provides the description of a system model. Section 4 describes the proposed security protocol. The simulation environment and experimental results are presented in Section 5. Finally, Section 6 concludes the paper.

2. Security Attacks in Mobile Ad Hoc Networks

Security means protecting the privacy (confidentiality), availability, integrity and non-repudiation. Security implies the identification of potential attacks from unauthorized access, use, modification or destruction. A *security attack* is any action that compromises the security of information in an unauthorized way. The attack may alter, release, or deny data [8][9][10]. The attacks on the MANETs can be broadly classified into two categories: passive and active attacks. Both passive and active attacks can be made on any layer of the network protocol stack [3].

2.1 Passive Attacks: A passive attack attempts to retrieve valuable information by listening to traffic channel without proper authorization, but does not affect system resources and the normal functioning of the network. Passive attacks are very hard to detect because they do not involve any alteration of the data. Figure 1 shows a schematic description of a passive attacker C, eavesdropping on the communication channel between A and B.

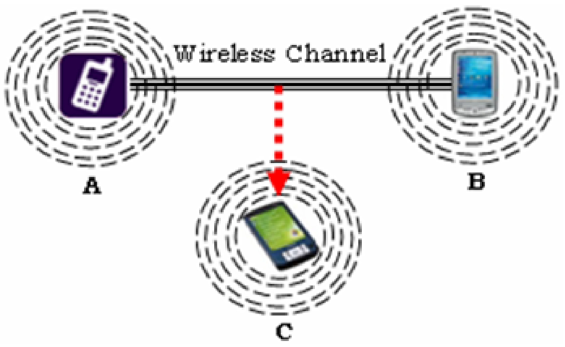


Figure 1: A passive attack

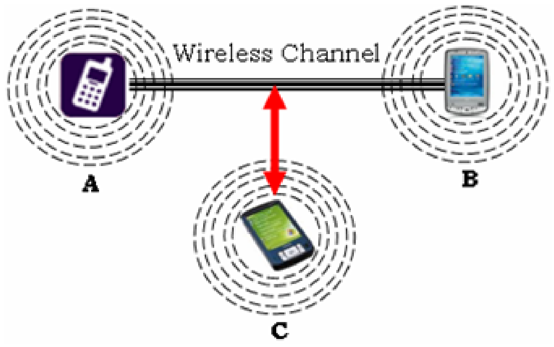


Figure 2: An active attack

2.2 Active Attacks: An active attack attempts to change or destroy the system resources. It gains an authentication and tries to affect or disrupt the normal functioning of the network services by injecting or modifying arbitrary packets of the data being exchanged in the network. An active attack involves information interruption, modification, or fabrication. As shown in Figure 2, an active attacker C can listen, modify, and inject messages into the communication channel between A and B.

Active attacks can be either *internal* or *external* [11]. External attacks are carried out by mobile nodes that do not fit into the network. These attacks are launched by adversaries who are not initially authorized to participate in the network operations and access the resources without authorization. Internal attacks are from cooperative mobile nodes that are part of the network.

Compared with external attacks, internal attacks are more serious and hard to detect because the attackers know valuable and secret information from compromised or hijacked nodes and possess privileged access rights to the network resources. Active attacks involve actions such as impersonation (masquerading or spoofing), modification, fabrication and replication. The active attacks are classified into different types as shown in Figure 3.

Jamming Attack - In this attack, the attacker primarily keeps examining the wireless medium to find out the frequency at which the receiver node is receiving signals from the dispatcher. It, then, transmits signals on that particular frequency so that free reception at the receiver is hindered without error [3].

Wormhole Attack - In this attack, a malicious node captures packets from one location in the network and "*tunnels*" these packets to the other malicious node at another location. The second malicious node is then expected to replay the "tunneled" packets locally. The tunnel between two colluding attackers is considered as a *wormhole*. The wormhole can drop packets by short-circuiting the regular flow of routing packets or it can carefully forward packets to avoid detection [12].

Black Hole Attack - This attack is a kind of denial of service where a malicious node attracts all packets by falsely claiming (advertising) a shortest path to the destination node whose packets it wants to intercept and, then, absorb them without forwarding to the destination [13].

Sinkhole Attack - In this attack, the adversary's goal is to attract all the virtual traffic from a specific area through a compromised node, creating a symbolic sinkhole with the opponent at the center as nodes on or near the path those packets follow have many opportunities to interfere with data [14].

Gray Hole Attack - A gray hole attack is a variation of the black hole attack, where the malicious node is not initially malicious, it turns malicious sometime later. In this attack, an attacker drops all data packets but it lets control messages to route through it [15].

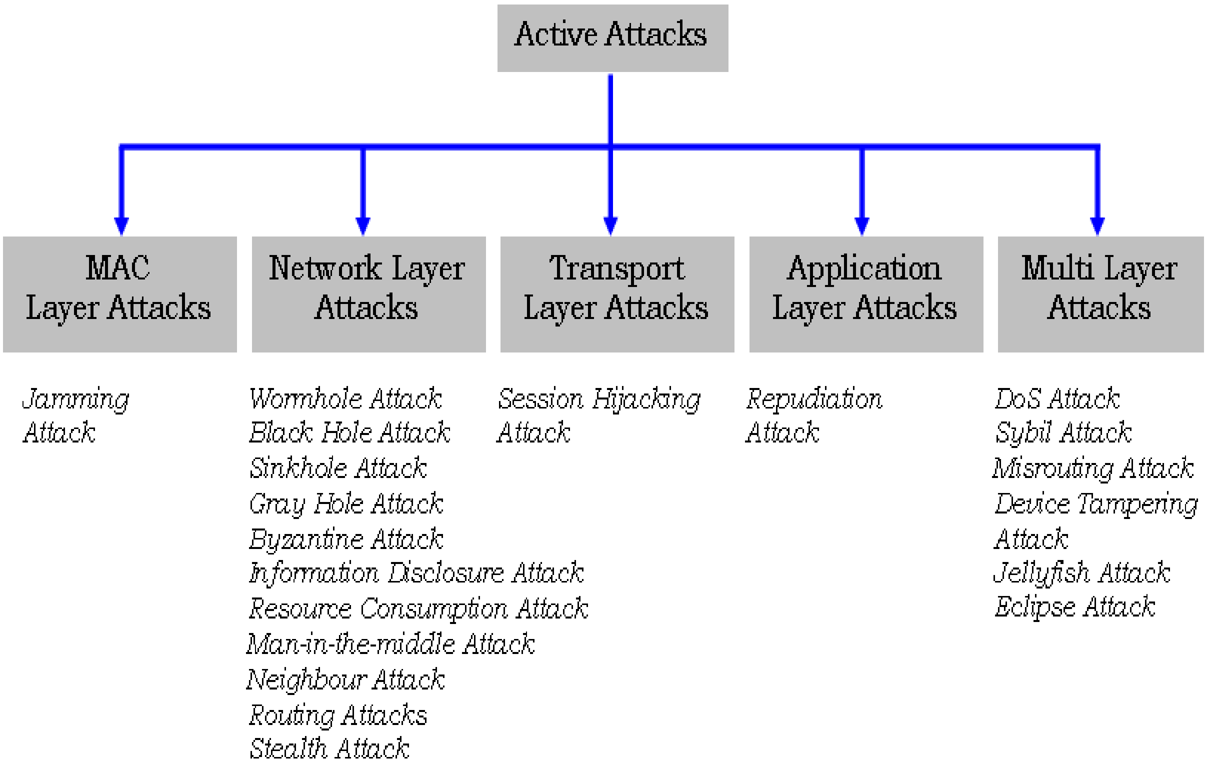


Figure 3: Classification of security attacks

Byzantine Attack - In this attack, a set of cooperative intermediate nodes works in combined and collectively performs attacks such as creating routing loops, routing packets on worst paths, and selectively dropping packets [16].

Information Disclosure Attack - In this, a compromised node attempts to reveal confidential or important information regarding the network topology, geographic locations of nodes, or optimal routes to unauthorized nodes in the network [5].

Resource Consumption Attack - In this attack, a malicious node intentionally tries to consume or misuse of the resources (battery power, bandwidth, and computational power) of other nodes' exist in the network by requesting excessive route discovery (unnecessary route request control messages), very frequent generation of beacon packets, or by forwarding unnecessary packets (stale information) to that node [3].

Man-In-The-Middle Attack - In this attack, the attacker exists as a neighbor to any one node in the routing path and alters data that is being transmitted and injects modified packet into network [10].

Neighbor Attack - The goal of neighbor attackers is to disrupt multicast routes by making two nodes that are in fact out of communication range believe that they can communicate directly with each other [13].

Routing Attacks - In this attack, attackers try to alter the routing information and data in the routing control packet. There are several types of routing attacks mounted on the routing protocol which are intended for disturbing the operation of the network [3].

Stealth Attacks - Stealth attacks are classified into two classes. The first class of attacks attempts to perform traffic analysis on filtered traffic to and from victim nodes. The second class partitions the network and reduces good put by disconnecting victim nodes in several ways. The methods are referred to as *stealth* attacks since they minimize the cost of launching the attacks [17].

Session Hijacking Attack – This attack is the major transport layer attack. Here, an adversary between two nodes takes control over a session. Once the session gets known between two nodes, the misbehaving node covers up as one of the end nodes of the session and takes control over the session [3].

Repudiation Attack - Repudiation attack is the main application layer level attack. Repudiation refers to the rejection or attempted denial by a node involved in a communication of having contributed in a part or the entire communication [3]. *Non-repudiation* is one of the key requirements for a security protocol in any communication network and assures that a node cannot later deny the data was sent by it.

Denial of Service Attack - In this attack, an adversary always attempts to avoid legitimate and authorized users of network services from accessing those services, where legitimate traffic cannot reach the target nodes [18].

Sybil Attack - This attack is also known as masquerade or impersonation or spoofing attack. In this attack, a single malicious node attempts to take out the identity of other nodes' in the network by advertising false/fake routes. It then attempts to send packets over network with identity of other nodes making the destination believe that the packet is from original source [19].

Misrouting Attack - This attack is also known as *manipulation of network traffic attack*. This is a very simple way for a node to disturb the protocol operation by announcing that it has better route than the existing one. In the misrouting attack, a on-legitimate node redirects the routing message and transfers data packet to the wrong target [20].

Device Tampering Attack - MANET nodes are generally compact, soft, and hand-held in nature. They might be broken or lost or stolen easily and misused by an opponent [3].

Jellyfish Attack - A jellyfish attacker first needs to intrude into the multicast forwarding group. It then interrupts data packets unreasonably for some time before forwarding them. This results high end-to-end delays and, thus, degrades the real-time applications performance [18].

Eclipse Attack - A pattern of misbehavior called an *eclipse* attack, which consists of the gradual poisoning of good (uncompromised) nodes' routing tables with links to a conspiracy of adversarial nodes (compromised nodes) [13].

3. System Model

Key management is the set of techniques and methods sustaining the setting up and maintenance of keying relationships between certified parties [8][9]. A hybrid key establishment scheme makes use of both symmetric and asymmetric techniques. The main problem with any public key cryptography based security system is to make each user's public key certificates available to others in such a way that its authenticity is verifiable.

Figure 4 shows the typical system model used in our work to develop the proposed scheme for secure routing. The network is drawn with 8 nodes. Each node contains the own public-private key pairs and own certificate, and constructs neighbor key repository (*NKR*), neighbor certificate repository (*NCR*), shared key repository (*SKR*) and trust graph (*TG*). The storage elements at each node are shown at node1. The certificate exchange packet is shown between node1 and node2. While the plaintext is in transit from node3 to node1 through the node2; node2 and node5 acts as wormhole attackers and creates the tunnel and node6 acts as man-in- middle attacker which corrupts the message. To avoid attacks, the message is transmitted in a secure way by encrypting the data. The encrypted data packet sent from node3 to node1 is shown at node3.

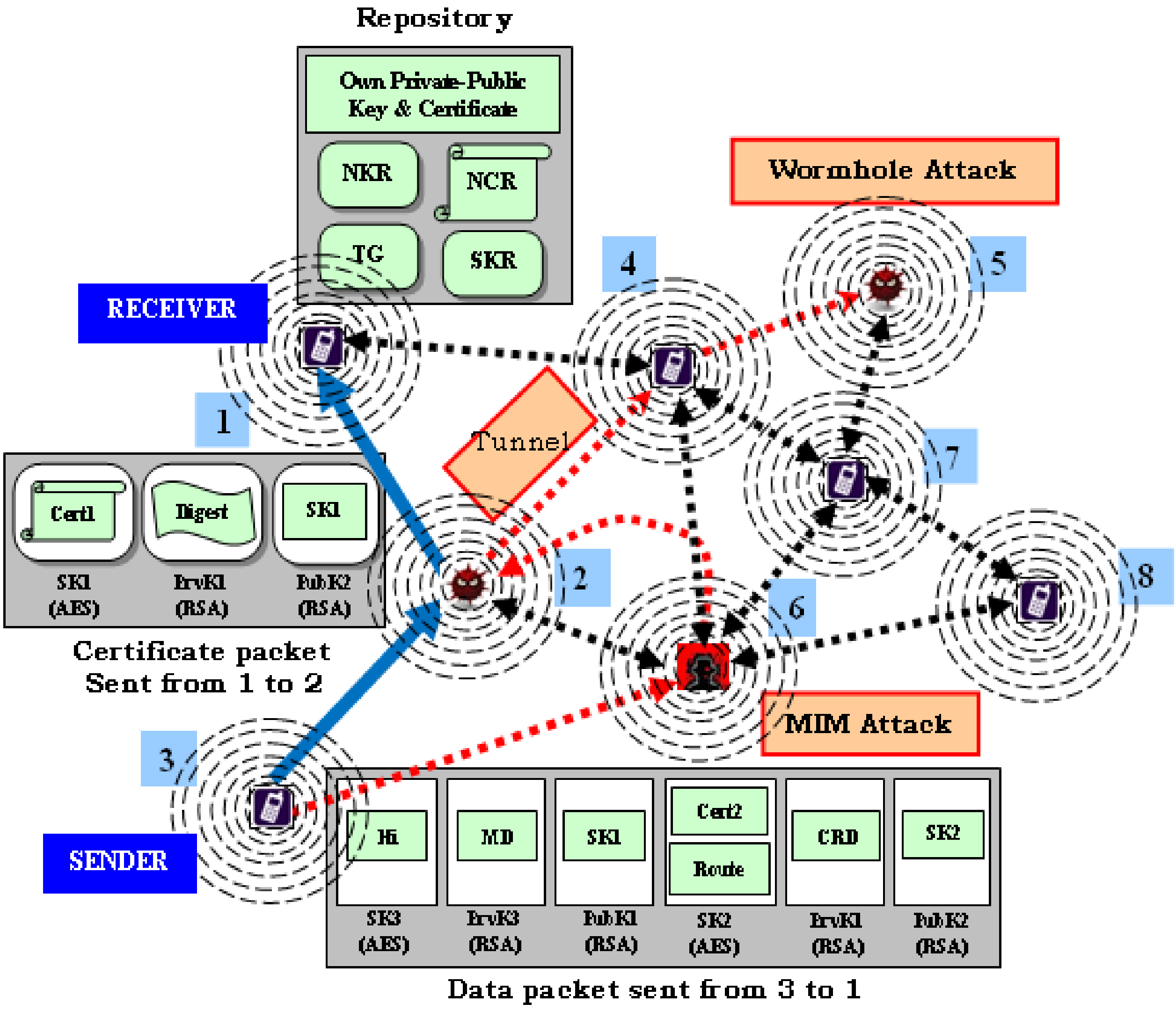


Figure 4: A system model for a new security protocol in MANET

4. A New Security Protocol

The proposed security protocol, called the *cryptographic hybrid key management for secure routing in MANETs*, provides the self-organized behavior by sharing the public keys and self-signed certificates among the nodes to form the network with an initial trust phase. The main goal of the proposed scheme is to provide a secure environment for transmission of messages from source to destination, where the source allows encrypting the messages that will be decrypted at destination only.

To secure a MANET, a security protocol must satisfy the attributes: confidentiality (privacy), availability, integrity, authenticity and non-repudiation. The proposed scheme achieves the confidentiality by encrypting the message with the sender's AES symmetric key generated for that message, thereby making it impossible for the attacker to get useful information from the data overheard. The receiver's RSA public key is used to encrypt the AES secret key. Then, the message digest is encrypted with the sender's RSA private key so that all the security goals are achieved. The algorithmic operations of the proposed security protocol are as follows:

Step1: //Each device creates public-private key pairs and public key certificates

Generate $K_S(X), K_P(X), C_X \quad \forall X \in N$; where N is number of devices, $K_S(X)$ is secret key of device X, $K_P(X)$ is public key of device X, and C_X is certificate of device X.

Initialize $NKR_X, NCR_X, SKR_X, SMR_X, RMR_X$; where NKR_X is neighbors key repository of device X, NCR_X is neighbors certificate repository of device X, SKR_X is shared key repository of device X, SMR_X is sent messages repository of device X , and RMR_X is receiving messages repository of device X.

Step2: //Exchange of public keys and construction of public key repository

Issue $K_P(X) \Rightarrow Z$; where $\forall Z \in NBR(X)$ and NBR(X) is neighbors of node X

Store $NKR_X \Leftarrow K_P(Z)$; // Receives $K_P(Z)$ and stores it in NKR_X

Step3: //Exchange of public key certificates and construction of certificate repository

$$PKT_X = (Enc_{AES}(C_X), Enc_{K_S(X)}(Digest(C_X)), Enc_{K_P(Z)}(AES));$$

where PKT_X is a data packet of device X, Enc_{Key} is encryption with the key.

Issue $PKT_X = Z$; $PKT_Z = (Dec_{K_S(X)}(AES), Dec_{AES}(C_Z), Dec_{K_P(Z)}(Digest(C_Z)));$

where Dec_{Key} is decryption with the key. Receives PKT_Z by device X and decrypt it and then verify the authentication.

if (decrypted value of $Digest(C_Z)$ $\equiv$ value of $Digest(C_Z)$ calculated at X) then $NCR_X \Leftarrow C_Z$

Step 4: //Each device constructs and exchanges of shared key repository and trust graph

$SKR_X \Leftarrow NKR_X$; Send $SKR_X \Rightarrow Z$; Receive SKR_Z and update SKR_X until SKR_X contains **N** public keys.

Construct TG_X and Send $TG_X \Rightarrow Z$; TGX is trust graph of node X. Receive TG_X and update TG_X ;

Step 5: //To find shortest route using fisheye state routing protocol

$$\text{if } ((SKR_S(D)\,\mathrm{Re}\,mainingKeyExpirationTime) > 60\,\mathrm{sec}\,onds\,)$$

$$\mathrm{Re}\,mainingKeyExpirationTime[X] = SKR_S(X)KeyGenerationTime - CurrentTime;$$

where $\forall S, D, X \in N$

// Initialization of Dijkstra’s variables

Initialize $State[X]$ with infinite length, label status as true; where $State[X]$ is the state of a device either *visited or not* while detecting the path.

$KeyExpirationTime[X] = Key\,\mathrm{Re}\,mainingExpirationTime[X];$

if $(X \equiv Y) adj[X][Y] = 0$; else { $adj[X][Y] = 1$; Count the total number of edges; where $\forall X, Y \in N$ }

$State[D].length = 0, State[D].status = false, Current = D$; where $\forall Current \in N$

repeat {

if $(adj[Current][X] \neq 0 andState[X])$ is not visited; where $\forall Current, X \in N$

{if ((State[Current].length + adj[Current][X] ≡ State[X].length) && KeyExpirationTime[Current] + KeyRemainingExpirationTime[X] >= KeyExpirationTime[X]) Assign label to State[X];

if (State[Current].length + adj[Current][X] < State[X].length) Assign label to State[X];

}

if ∃X with minimum length, then Current = X;

} until source node is detected/found;

repeat { paths$_S$(D) = State[X].next; } until destination is reached;

if determined path contains D then return Paths$_S$(D), where $\forall D \in Path_S(D)$;

Step 6: //The plain text transmission from a given source to destination

$SMR_S \Leftarrow (M_{S,D}, C_Z \oplus Path_S(D))$; where $\forall S, D, Z \in N$, Paths$_S$(D) is the shortest path from source(S) to destination (D), $M_{S,D}$ is a message sent from source(S) to destination(D), and SMR_S is the sent message repository at source.

$PKT_S \Leftarrow (M_{S,D}, C_Z \oplus Path_S(D))$; where PKT_S is a data packet at the source and $\forall S, D, Z \in Path_S(D)$

Send $PKT_S \Rightarrow Z$ which in turn forwards $PKT_Z \Rightarrow Y$ until it reaches the destination, where $\forall S, Z, Y \in Path_S(D)$

$RMR_D \Leftarrow (M_{S,D}, C_D \oplus Path_S(D))$;

where $\forall S, D \in Path_S(D)$ and RMR$_D$ is received message repository at destination

Step 7: //Behavior of security attacks

if (*device X is a wormhole attacker*), creates tunnel path and send PKT$_X$ ⇒ Y; which in turn forwards PKT$_Y$ ⇒ W until it reaches to another attacker, where tunnel path$_{W1}$(W2) is a tunnel path created between wormhole1 to wormhole2, $\forall X \in Path_S(D)$, $\forall Y, W \in tunnelpath_X(Z)$;

if (*device X is a man-in-the-middle attacker*), PKT$_X$ = (M$_{S,\ D}$ +"Invalid random number") corrupts data and forwards PKT$_X$ ⇒ Y, where $\forall X \in N$, $\forall X \in NBR(Y)$, $\forall Y \in Path_S(D)$;

if (*device X is a denial of service attacker*) No forwarding of data, where $\forall X \in Path_S(D)$;

if (*device X is a sybil attacker*) Send invalid P$_X$ with spoofed ID's of Z where $\forall X \in N$, $\forall Z \in NBR(X)$;

if (*external misbehavior attacker enters into the network and try for connection establishment* *within the devices radio range*) then Raises Authentication error “Authentication Failed”;

Step 8: //Secure plain text transmission from source to destination

$SMR_S \Leftarrow (M_{S,D}, C_Z \oplus Path_S(D))$; where $\forall S,D,Z \in N$, Paths$_S$(D) is the shortest path from source(S) to destination (D), $M_{S,D}$ is a message sent from source(S) to destination(D), and SMR_S is the sent message repository at source.

$$\begin{aligned}&PKT_S = (Enc_{AES1}(M_{S,D}), Enc_{K_S(S)}(Digest(M_{S,D})), Enc_{K_P(D)}(AES1),\\ &Enc_{AES2}(C_X \oplus Path_S(D)),\\ &Enc_{K_S(S)}(Digest(C_X)), Enc_{K_P(X)}(AES2))\end{aligned}$$

$$\forall S, X \in Path_S(D), \forall X \in NBR(S)$$

Send $PKT_S \Rightarrow X$ which in turn sends $PKT_X \Rightarrow Z$, where $\forall S, X, Z \in Path_S(D), \forall X \in NBR(S)$

// Certificate is decrypted at each hop and PKT_Z is forwarded until it reaches the destination.

$\begin{aligned}&\mathrm{PKT_D} \Leftarrow (Dec_{K_S(D)}(AES2),\\ &Dec_{AES2}(C_D \oplus Path_S(D)), Dec_{K_P(Z)}(Digest(C_D)))\end{aligned}$; where PKT_D is a packet decryption at destination and $\forall D \in Path_S(D), \forall Z \in NBR(D)$

if (decrypted value of $digest(C_D)$ ≡ values of $digest(C_D)$ calculated at D) then

$\begin{aligned}&PKT_D \Leftarrow (Dec_{K_S(D)}(AES1), Dec_{AES1}(M_{S,D}),\\ &Dec_{K_P(S)}(Digest(M_{S,D})))\end{aligned}$ // Neighbor authentication

if (decrypted value of $digest(M_{S,D})$ ≡ value of $digest(M_{S,D})$ calculated at D) then

$RMR_D \Leftarrow (M_{S,D}, C_D \oplus Path_S(D))$; //Source authentication where $\forall S, D \in Path_S(D)$ and RMR$_D$ is received message repository at destination.

5. Experimental Results and Analysis

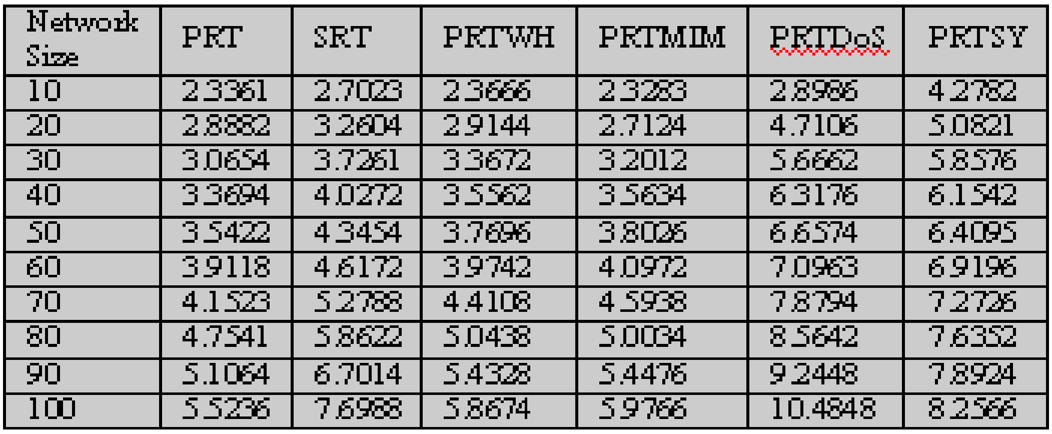
This section describes the experimental network scenarios and the analysis of simulation results. We analyze the security of a proposed security protocol via the impact of different types of security attacks on secure routing. The proposed scheme has been implemented in Java SE 6 with lightweight Bouncy Castle 1.6 API. Simulation results have shown that the proposed scheme resists against malicious nodes with low implementation complexity.

The security protocol solutions, proposed in the present work, rely on security mechanisms - private and public key cryptography (AES, RSA, X.509 certificates, digital signatures) and secure hash based message authentication codes (SHA1). The use of cryptographic principles takes more time to encrypt and decrypt at every node. To avoid this, we have used the hybrid encryption techniques both the symmetric and asymmetric algorithms.

First, we study the impact of various security attacks against the plain message transmission from source to destination. Next, we study the routing overhead against the secure message transmission from source to destination.

Man-in-the-middle attack - As shown in Figure 5, since all the messages are signed, the attacker has no other choice than to use his own private key and the corresponding public key as the identifier. *Sybil attack* - As shown in Figure 6, the fact that each message carries a digital signature, and that a node’s identity is bound to its public key, make impersonation attacks impossible. *Wormhole attack* - As shown in Figure 7, the fact that each message that is sent over the network is encrypted and each message carries digital signature makes and wormhole attacks impossible because the symmetric key which is used for encrypting the message is encrypted with the public key of destination. Hence, the destination node can only decrypt the symmetric key which is, now, used to decrypt the message.

The comparisons are made on routing overhead against the security attacks. Table 1 shows the data routing time, in seconds, for different network sizes. Each data point in the resulting table is an average of four program runs with an identical configuration of various network sizes, but different randomly generated mobility patterns.



Network Size	PRT	SRT	PRTWH	PRTMIM	PRTDoS	PRTSY
10	2.3361	2.7023	2.3666	2.3283	2.8986	4.2782
20	2.8882	3.2604	2.9144	2.7124	4.7106	5.0821
30	3.0654	3.7261	3.3672	3.2012	5.6662	5.8576
40	3.3694	4.0272	3.5562	3.5634	6.3176	6.1542
50	3.5422	4.3454	3.7696	3.8026	6.6574	6.4095
60	3.9118	4.6172	3.9742	4.0972	7.0963	6.9196
70	4.1523	5.2788	4.4108	4.5938	7.8794	7.2726
80	4.7541	5.8622	5.0438	5.0034	8.5642	7.6352
90	5.1064	6.7014	5.4328	5.4476	9.2448	7.8924
100	5.5236	7.6988	5.8674	5.9766	10.4848	8.2566

Table 1: Comparison of routing overhead against security attacks

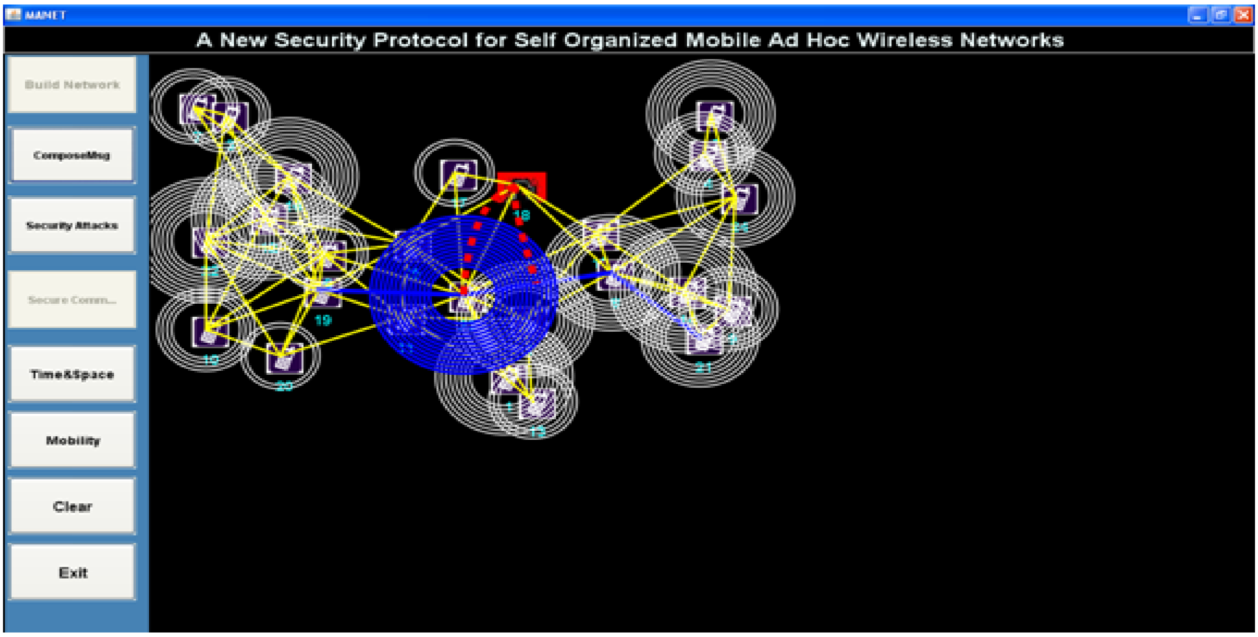


Figure 5: A snapshot of showing a *man-in-the-middle attacker* at node18. It is consuming data from node23 and injecting modified data into network

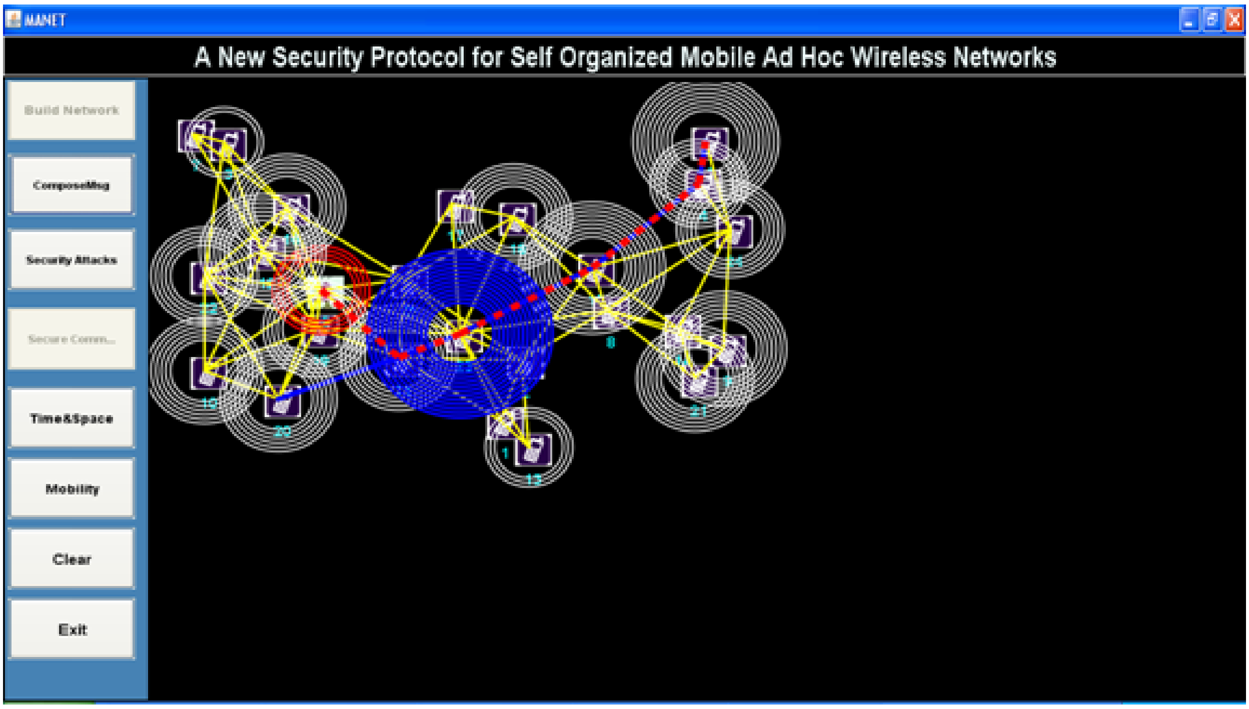
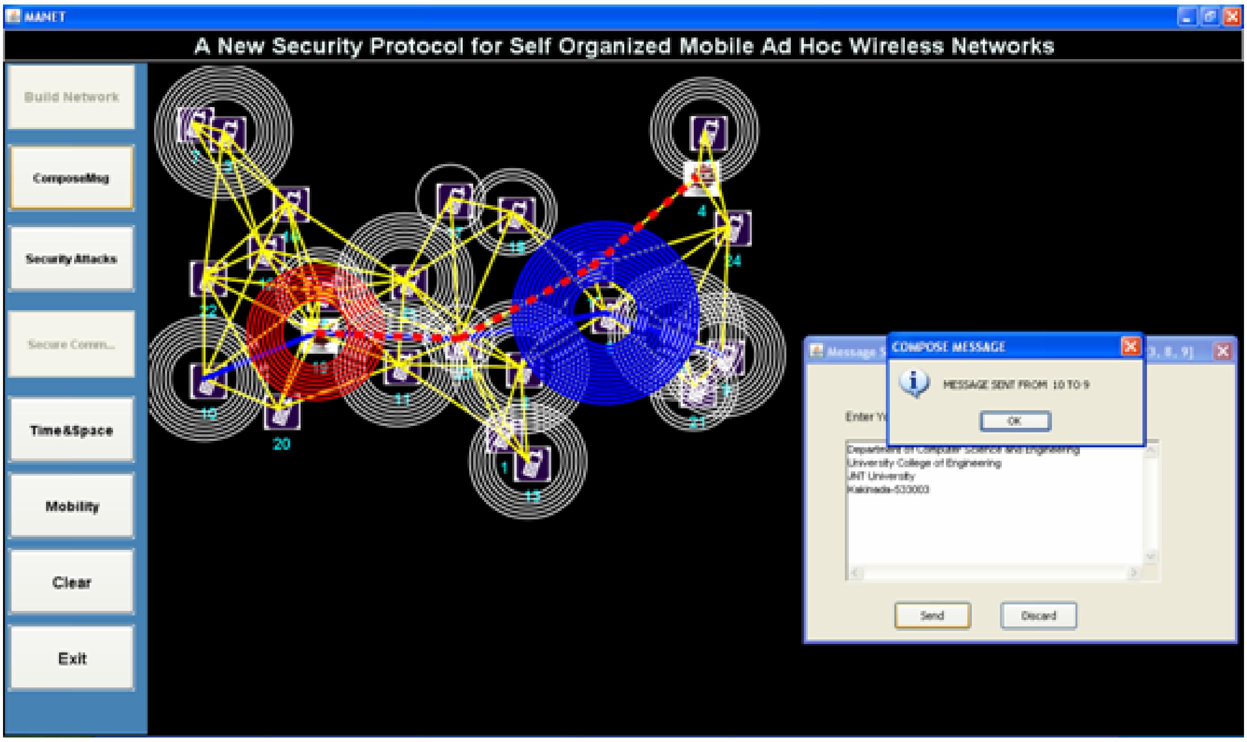


Figure 6: A snapshot of showing a *Sybil attacker* spoofing the identities of other nodes and forwarding data to destination as if the actual node is forwarding data



MESSAGE SENT FROM 10 TO 9

Department of Computer Science and Engineering
University College of Engineering
JNT University
Kakinada-533003

Figure 7: A snapshot of showing a *wormhole attacker* at node19 tunneling the packets to other attacker at node4 instead of destination

When the network size is increased to 100, the various routing times (in seconds) are: Plaintext Routing Time (*PRT*): 5.5236,

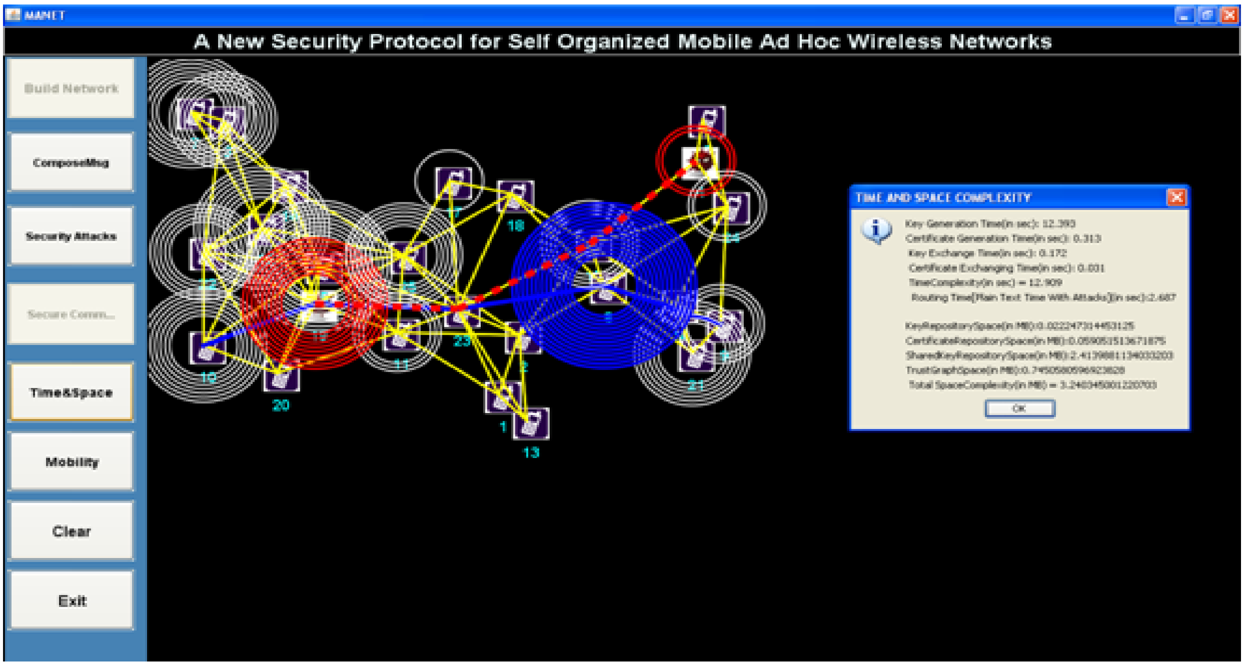
Secure text Routing Time (*SRT*): 7.6988,

Plaintext Routing Time with *Wormhole Attack* (*PRTWH*): 5.8674,

Plaintext Routing Time with *Man-In-The-Middle Attack* (*PRTMIM*): 5.9766,

Plaintext Routing Time with *DoS Attack* (*PRTDoS*): 10.4848, and

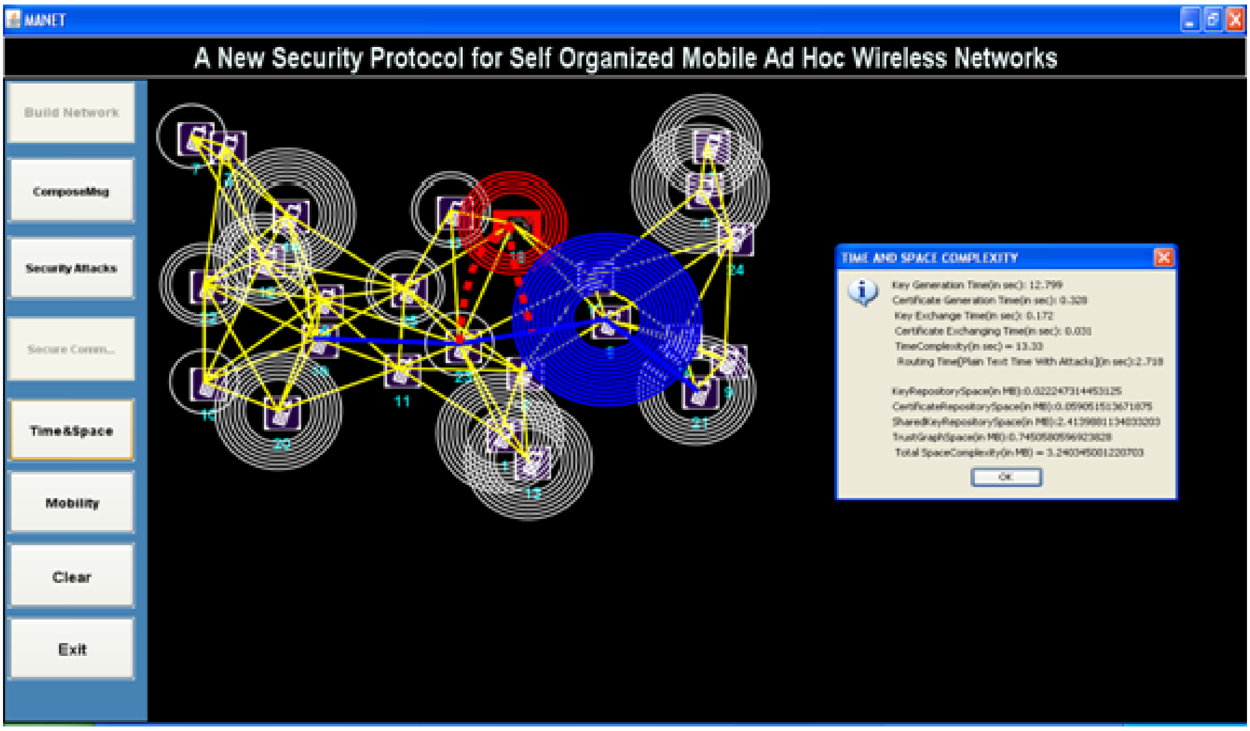
Plaintext Routing Time with *Sybil Attack (PRTSY)*: 8.2566.



```
Key Generation Time(in sec): 12.393
Certificate Generation Time(in sec): 0.313
 Key Exchange Time(in sec): 0.172
 Certificate Exchanging Time(in sec): 0.031
 TimeComplexity(in sec) = 12.909
  Routing Time[Plain Text Time With Attacks]](in sec):2.687

KeyRepositorySpace(in MB):0.0222473144453125
CertificateRepositorySpace(in MB):0.059051513671875
SharedKeyRepositorySpace(in MB):2.4139881134033203
TrustGraphSpace(in MB):0.7450580596923828
 Total SpaceComplexity(in MB) = 3.240345001220703
```

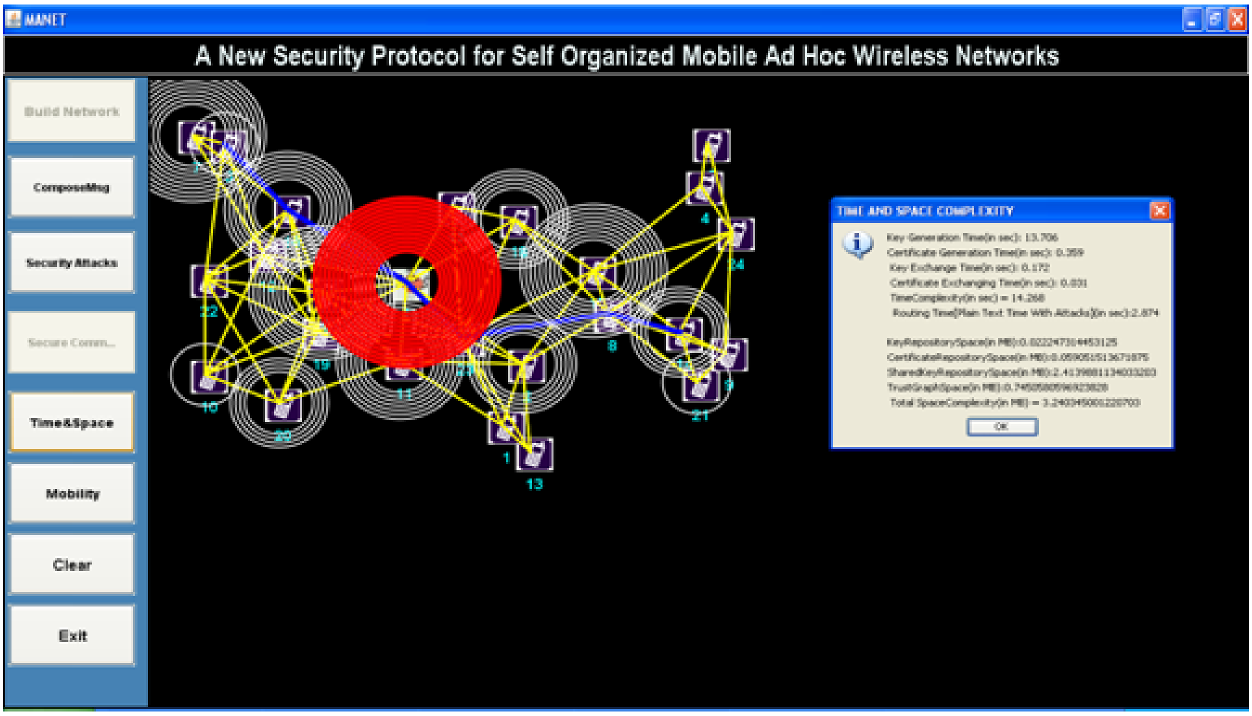
Figure 8: A snapshot of showing a routing time in case of *wormhole* attacker of network size 25 nodes


MANET
A New Security Protocol for Self Organized Mobile Ad Hoc Wireless Networks
Build Network
ComposeMsg
Security Attacks
Secure Comm...
Time&Space
Mobility
Clear
Exit
24
11
21
23
TIME AND SPACE COMPLEXITY
OK

```
Key Generation Time(in sec): 12.799
Certificate Generation Time(in sec): 0.328
 Key Exchange Time(in sec): 0.172
 Certificate Exchanging Time(in sec): 0.031
 TimeComplexity(in sec) = 13.33
  Routing Time[Plain Text Time With Attacks]](in sec):2.718

KeyRepositorySpace(in MB):0.0222473144453125
CertificateRepositorySpace(in MB):0.059051513671875
SharedKeyRepositorySpace(in MB):2.4139881134033203
TrustGraphSpace(in MB):0.7450580596923828
 Total SpaceComplexity(in MB) = 3.240345001220703
```

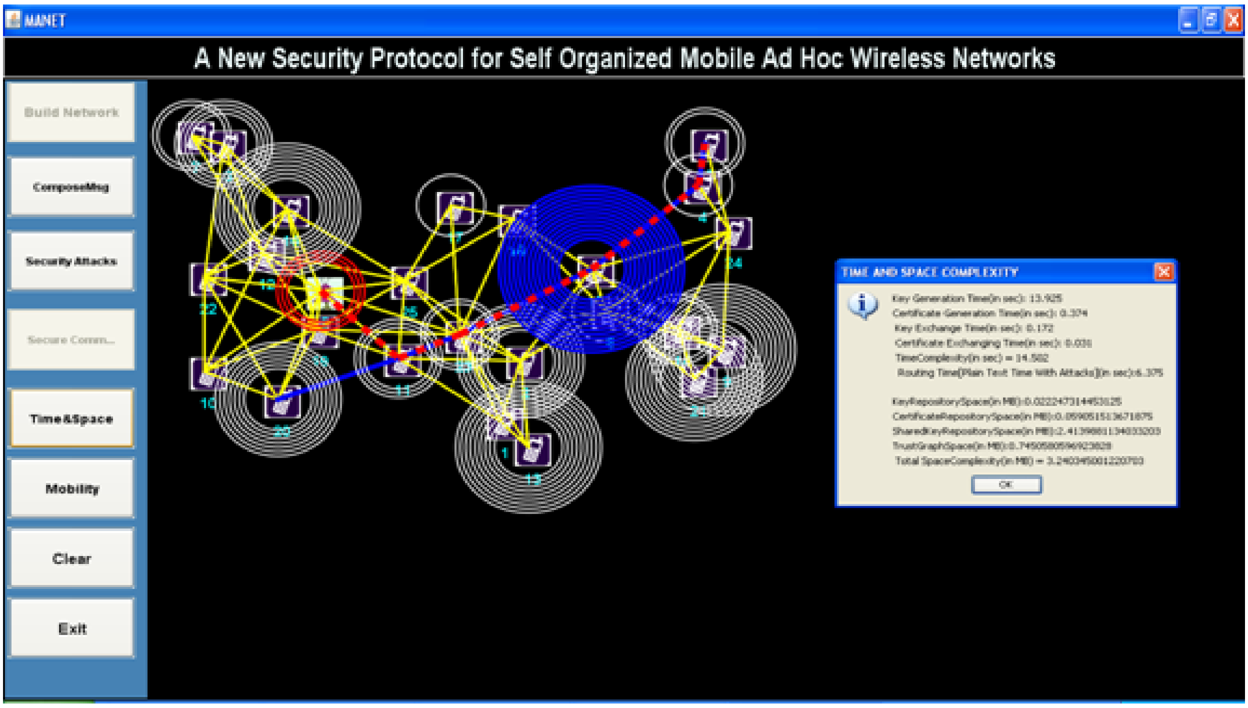
Figure 9: A snapshot of showing a routing time in case of *man-in-the-middle* attacker of network size 25 nodes


MANET
A New Security Protocol for Self Organized Mobile Ad Hoc Wireless Networks
Build Network
ComposeMsg
Security Attacks
Secure Comm...
Time&Space
Mobility
Clear
Exit
TIME AND SPACE COMPLEXITY
OK

```
Key Generation Time(in sec): 13.706
Certificate Generation Time(in sec): 0.359
 Key Exchange Time(in sec): 0.172
 Certificate Exchanging Time(in sec): 0.031
 TimeComplexity(in sec) = 14.268
  Routing Time[Plain Text Time With Attacks](in sec):2.874

KeyRepositorySpace(in MB):0.022247314453125
CertificateRepositorySpace(in MB):0.059051513671875
SharedKeyRepositorySpace(in MB):2.413988113403203
TrustGraphSpace(in MB):0.7450580596923828
 Total SpaceComplexity(in MB) = 3.240345001220703
```

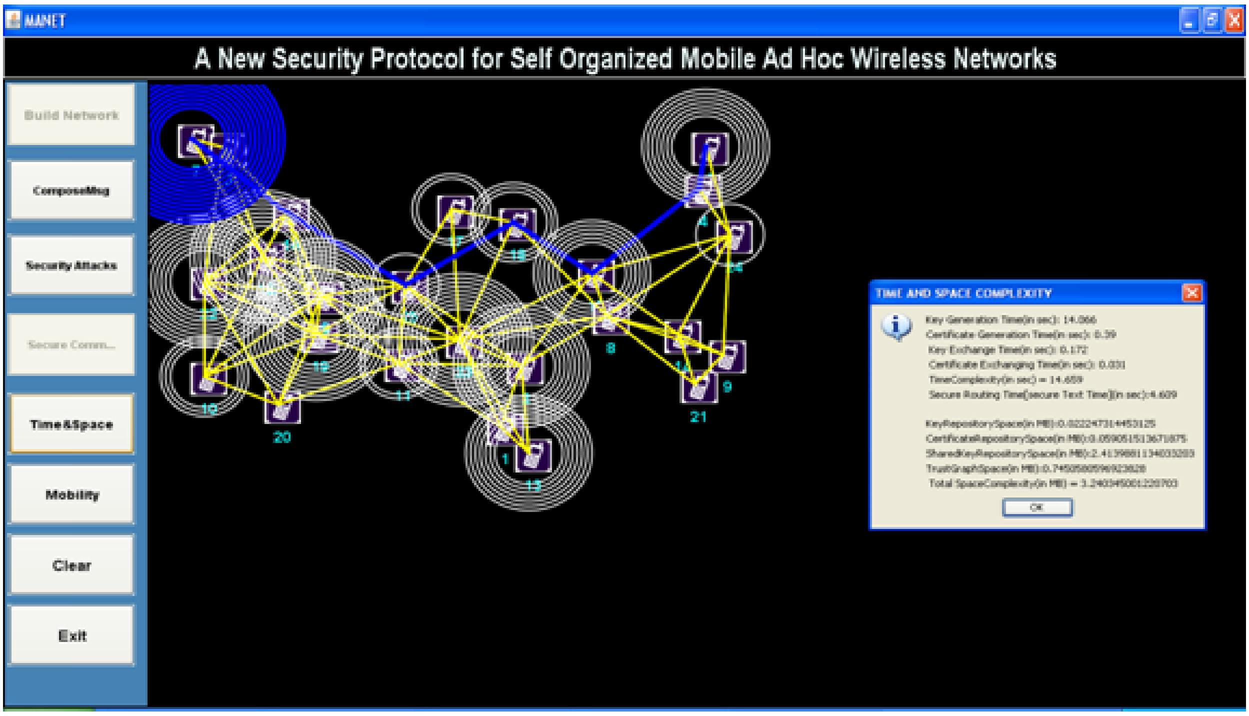
Figure 10: A snapshot of showing a routing time in case of *Denial of Service* attacker of network size 25 nodes


MANET
A New Security Protocol for Self Organized Mobile Ad Hoc Wireless Networks
Build Network
ComposeMsg
Security Attacks
Secure Comm...
Time&Space
Mobility
Clear
Exit
TIME AND SPACE COMPLEXITY
OK

```
Key Generation Time(in sec): 13.925
Certificate Generation Time(in sec): 0.374
 Key Exchange Time(in sec): 0.172
 Certificate Exchanging Time(in sec): 0.031
 TimeComplexity(in sec) = 14.502
  Routing Time[Plain Text Time With Attacks](in sec):6.375

KeyRepositorySpace(in MB):0.022247314453125
CertificateRepositorySpace(in MB):0.059051513671875
SharedKeyRepositorySpace(in MB):2.413988113403203
TrustGraphSpace(in MB):0.7450580596923828
 Total SpaceComplexity(in MB) = 3.240345001220703
```

Figure 11: A snapshot of showing a routing time in case of *Sybil* attacker of network size 25 nodes


MANET
A New Security Protocol for Self Organized Mobile Ad Hoc Wireless Networks
Build Network
ComposeMsg
Security Attacks
Secure Comm...
Time&Space
Mobility
Clear
Exit
TIME AND SPACE COMPLEXITY
OK

Key Generation Time(in sec): 14.066
Certificate Generation Time(in sec): 0.39
Key Exchange Time(in sec): 0.172
Certificate Exchanging Time(in sec): 0.031
TimeComplexity(in sec) = 14.659
Secure Routing Time[secure Text Time](in sec):4.609

KeyRepositorySpace(in MB):0.022247314453125
CertificateRepositorySpace(in MB):0.059051513671875
SharedKeyRepositorySpace(in MB):2.4139881134033203
TrustGraphSpace(in MB):0.7450580596923828
Total SpaceComplexity(in MB) = 3.240345001220703

Figure 12: A snapshot of showing a routing time while secure text transmission of network size 25 nodes

The resulting data of the table 1 are plotted using MATLAB 7.6 [25] and is shown in Figure 13.

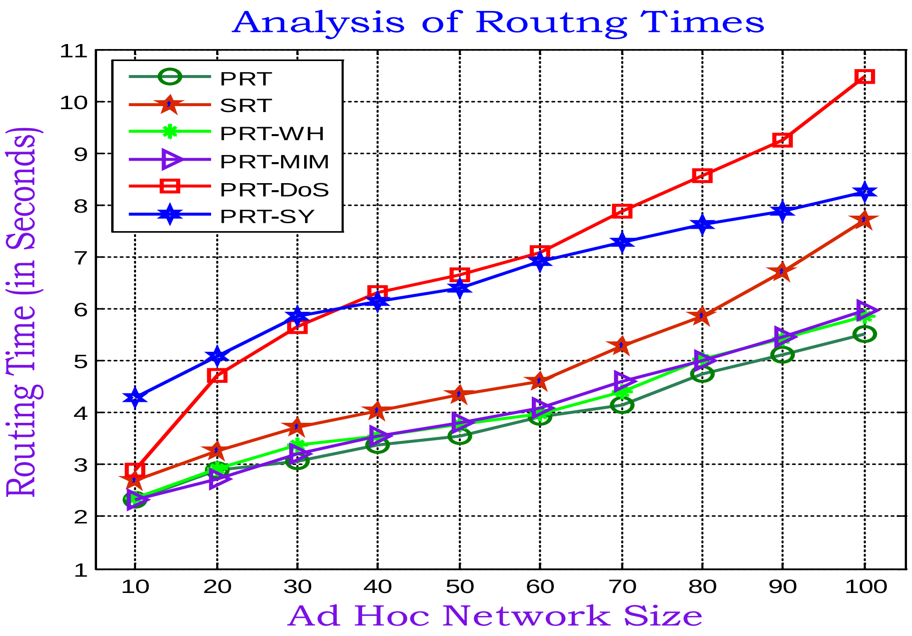


Figure 13: Analysis of routing overhead with and without security attacks

6. Conclusion

The proposed security protocol has been implemented in Java SE 6 with lightweight

Bouncy Castle 1.6 API and empirically evaluated its performance via the impact of different types of security attacks and simulation assessments. The comparisons are made on routing overhead against the security attacks. Certificate successful rate is better when the numbers of malicious nodes are increased over the different network sizes. Simulation results have shown that the proposed scheme resists against malicious nodes with low implementation complexity. It has been found that the proposed approach is an effective way of providing security in MANETs.

7. References

[1] C.K. Tok, “Ad Hoc Mobile Wireless Networks: Protocols and Systems”, Pearson Education, pp. 28-30, 2002.

[2] X. Cheng, X. Huang and D. Z Du, “Ad Hoc Wireless Networking”, Kluwer Academic Publishers, ISBN: 1-4020-7712-2, pp. 319-364, 2006.

[3] C. Siva Ram Murthy and B.S Manoj, “Ad Hoc Wireless Networks: Architectures and Protocols”, Pearson Education, ISBN: 978-81-317-0688-6, 2006.

[4] F. Anjum and P. Mouchtaris, “Security for Wireless Ad Hoc Networks”, John Wiley & Sons, 2007.

[5] S. Basagni, M. Conti, S. Giordono and I. Stojmenovic, “Mobile Ad Hoc Networking”, IEEE Press, John Wiley & Sons, New York, 2004.

[6] S. Misra, I. Woungang and S. C. Misra, “Guide to Wireless Ad Hoc Networks”, Springer, Berlin, 2009.

[7] C. M. Cordeiro and D. P. Agrawal, “Ad Hoc & Sensor Networks: Theory and Applications”, World Scientific Publishing, Singapore, 2006.

[8] W. Stallings, “Cryptography and Network Security: Principles and Practice”, Fifth Edition, Prentice Hall, 2010.

[9] A. J. Menezes, P. C. V. Oorschot, and S. A. Vanstone, “Handbook of Applied Cryptography”, CRC Press, ISBN: 0849385237, 1996.

[10] C. Gandhi and M. Dave, “A Review of Security in Mobile Ad Hoc Networks”, IETE Technical Review, ISSN: 02564602, pp.335-344, Vol. 23, No. 6, 2006. http://dx.doi.org/10.1080/02564602.2006.11657964

[11] Prasant Mohapatra and Srikanth V. Krishnamurthy, “Ad Hoc Networks: Technologies and Protocols”, Springer International Edition, 2005.

[12] Yi-Chun Hu, A. Perrig, and D. B. Johnson, “Wormhole Attacks in Wireless Networks”, IEEE Journal on Selected Areas in Communications, pp.370-380, Vol.24, No.2, 2006, http://dx.doi.org/10.1109/JSAC.2005.861394

[13] Yi-Chun Hu and Adrian Perrig, “A Survey of Secure Wireless Ad Hoc Routing”, IEEE Security and Privacy, pp. 28-39, Vol.2, No.3, 2004, http://dx.doi.org/10.1109/MSP.2004.1

[14] Yih-Chun Hu, Adrian Perrig, and David B. Johnson, “Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks”, Proceedings of 8th ACM International Conference on Mobile Computing and Networking (MobiCom-2002), pp.12-23, Atlanta, GA, USA, 2002. http://dx.doi.org/10.1007/s11276-004-4744-y

[15] Jaydip Sen, M. Girish Chandra, Harihara S.G., Harish Reddy and P. Balamuralidhar, “A

Mechanism for Detection of Gray Hole Attack in Mobile Ad Hoc Networks", Proceedings of IEEE 6th International Conference on Information, Communications and Signal Processing, pp.1-5, 2007. http://dx.doi.org/10.1109/ICICS.2007.4449664

[16]B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An On-Demand Secure Routing Protocol Resilient to Byzantine failures", Proceedings of 1st ACM Workshop on Wireless Security (WiSe'02), ISBN: 1-58113-585-8, pp.10, NY, USA, 2002. http://dx.doi.org/10.1145/570681.570684

[17]M. Jakobsson, S. Wetzel and B. Yener, "Stealth Attacks on Ad Hoc Wireless Networks", Proceedings of IEEE 58th Vehicular Technology Conference, pp.2103-2111, Vol.3, 2003. http://dx.doi.org/10.1109/VETECF.2003.1285396

[18] I. Aad, J.P. Hubaux, and E.W. Knightly, "Denial of Service Resilience in Ad Hoc Networks", Proceedings of the ACM 10th Annual International Conference (MobiCom-2004), Philadelphia, PA, 2004, http://dx.doi.org/10.1145/1023720.1023741

[19]J. Douceur, "The Sybil Attack", Proceedings of the 1st International Workshop on Peer-to- Peer Systems (IPTPS'02), pp.251-260, Cambridge, MA, 2002.

[20]K. Sanzgiri, B. Dahill, B.N. Levine, C. Shields and E. M. Belding-Royer, "A Secure Routing Protocol for Ad hoc Networks", Proceedings of 10th IEEE International Conference on Network Protocols (ICNP2002), pp.78-87, Paris, France, November 2002.

[21] S. Capkun, L. Buttyan, and J. P. Hubaux, "Self-Organized Public-Key Management for Mobile Ad Hoc Networks", IEEE Transactions on Mobile Computing,Vol.2, No.1, pp.52–64, 2003, http://dx.doi.org/10.1109/TMC.2003.1195151

[22] S. Capkun, J. P. Hubaux, and L. Buttyan, "Mobile Helps Peer-to-Peer Security", IEEE Transactions on Mobile Computing, Vol. 5, No. 1, pp.43–51, 2006, http://dx.doi.org/10.1109/TMC.2006.12

[23]Daeseon CHOI, Younho LEE, Yongsu PARK, Seung-hun JIN, and Hyunsoo YOON, "Efficient and Secure Self-Organized Public Key Management for Mobile Ad Hoc Networks", IEICE Transactions on Communications, Vol. E91– B, No.11, pp. 3574-3583, 2008. http://dx.doi.org/10.1093/ietcom/e91-b.11.3574

Copyright Disclaimer