A NOVEL APPROACH FOR DETECTING WORMHOLE ATTACK USING RC4 ALGORITHM IN MANET

A PROJECT REPORT

Submitted by

S.PRINCE SAHAYA BRIGHTY

in partial fulfilment for the requirement of award of the degree of

MASTER OF ENGINEERING

in

COMPUTER SCIENCE AND ENGINEERING

Department of Computer Science and Engineering

KUMARAGURU COLLEGE OF TECHNOLOGY, COIMBATORE 641 049

(An Autonomous Institution Affiliated to Anna University, Chennai)

APRIL 2013

BONAFIDE CERTIFICATE

Certified that this project work titled **"A NOVEL APPROACH FOR DETECTING WORMHOLE ATTACK USING RC4 ALGORITHM IN MANET"** is the bona fide work of Ms. S.PRINCE SAHAYA BRIGHTY, who carried out the research under my supervision. Certified further, that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other students.

(Signature of the HoD) **Prof.N.JAYAPATHI M.Tech.,**	(Signature of the Supervisor) **D.SATHYA M.E.,**
HEAD OF THE DEPARTMENT	**SUPERVISOR**
Professor, Dept. of Computer Science and Engineering Kumaraguru College of Technology Coimbatore-641049	Assistant Professor, Dept.of Computer Science and Engineering Kumaraguru College of Technology Coimbatore-641049

Submitted for the Project Viva-Voce examination held on ____________.

--------------------------------	-------------------------------
Internal Examiner	**External Examiner**

ஆய்வுச்சுருக்கம்

இடம் பெயர்ந்து செல்கிற வலையமைப்பு அதாவது இழுது என்பது கம்பியில்லா இணைப்புகளால் இணைக்கப்பட்ட இடம் பெயர்ந்து செல்கிற சாதனங்களின் ஒரு சுய கட்டமைப்பு பிணையமாக உள்ளது. இதிலுள்ள ஒவ்வொரு சாதனமும் சுதந்திரமாக எந்த திசையிலும் நகரும். எனவே அதன் தொடர்பு அவ்வப்போது மாறும். இந்த இடம் பெயர்ந்து செல்கிற வலையமைப்பு பல தாக்குதல்களுக்கு உள்ளாக வாய்ப்புள்ளது. இத்தகைய வலையமைப்பின் பாதுகாப்பை அதிகரிக்க ஒரு முயற்சியாக பரவெளி அனுமான இணைப்பின் தாக்குதலை அடையாளம் காண இணைப்பு தகவலை பயன்படுத்த முன்மொழியப்பட்டது. முன்மொழியப்பட்ட நுட்பம் பரவெளி அனுமான தாக்கத்தை கண்டறிய அண்டை இடையேயான தகவல் வழி வேறுபாட்டினை பயன்படுத்துகிறது. விளக்கவுரையின் அடிப்படை என்னவென்றால் அனுப்புனரிடம் இருந்து துவங்கி இரண்டாவது தத்தித் தாண்டும் மாற்று பாதையை கண்டறியவும் மற்றும் பரவெளி அனுமான இணைப்பின் தாக்கத்தை கண்டறிய தத்தித் தாண்டும் மொத்த எண்ணிக்கையும் கணக்கிட உள்ளது.

ABSTRACT

Mobile Adhoc Network (MANET) is a self-configuring infrastructure less network of mobile devices connected by wireless links. Each device in an adhoc network is free to move independently in any direction, and hence its link gets changed frequently. Wireless adhoc network is prone to many attacks. In an attempt to enhance security in Manet, unobservable routing protocol focuses on wormhole attack. A novel algorithm was proposed that uses connectivity information to identify the wormhole attack. The proposed technique makes use of variance in routing information between neighbors to detect wormholes. The base of dissertation is to find alternate path from source to second hop and calculate the number of hops to detect the wormhole. The security routing protocol was implemented on Java and RC4 algorithm has satisfactory performance in terms of packet delivery ratio, latency and normalized control bytes.

ACKNOWLEDGEMENT

I express my profound gratitude to **Padmabhusan Arutselvar Dr.N.Mahalingam, B.Sc., F.I.E , Chairman**, **Dr.B.K. Krishnaraj Vanavarayar**, **Co-Chairman, Mr. M. Balasubramaniam, M.Com., M.B.A, Correspondent, Mr.Sankar Vanavarayar**, **M.B.A., PGDIEM, Joint Correspondent** and **Dr.S.Ramachandran Ph.D., Principal** for providing the necessary facilities to complete my thesis.

I take this opportunity to thank **Prof.N.Jayapathi M.Tech.,** Head of the Department, Department of Computer Science and Engineering, for his support and timely motivation. Special thanks to my Project Coordinator **Dr.V.Vanitha M.E., Ph.D.**, Senior Associate Professor, Department of Computer Science and Engineering, and project committee members for arranging brain storming project review sessions.

I register my sincere thanks to my Guide **Ms.D.Sathya M.E.,** Assistant Professor, Department of Computer Science and Engineering. I am grateful for her support, encouragement and ideas. I would like to convey my honest thanks to all **Teaching** and **Non Teaching Staff** members of the department and my classmates for their support.

I dedicate this project work to my **Parents** for no reasons but feeling from bottom of my heart, without their love this work would not be possible.

-S.PRINCE SAHAYA BRIGHTY

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

GPS	Geographical Positioning System
IOI	Item of Interest
MANET	Mobile Adhoc NETwork
RC4	Rivest Cipher 4
RREP	Route Reply
RREQ	Route Request
USOR	Unobservable Secure On-demand Routing Protocol

CHAPTER 1

INTRODUCTION

Privacy protection of the mobile adhoc network is more demanding than that of wired networks due to the open nature and mobility of wireless media. The basic architecture of the mobile adhoc network is shown in Fig.1.1. In Wireless Network, the attacker only needs an appropriate transceiver to receive wireless signals without being detected. In wired networks, devices like desktops are always static and do not move from one place to another. Hence in wired networks there is no need to protect users mobility behavior or movement pattern. This information should be kept private from adversaries in wireless environments. Otherwise, an adversary is able to profile users according to their behaviors. Providing privacy protection for adhoc networks with low-power wireless devices and low-bandwidth network connection is a very challenging task.

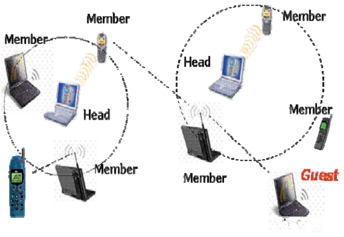


Fig.1.1. MANET Architecture

1.1 AD-HOC ROUTING PROTOCOLS

The mobile adhoc network does not rely upon any fixed support infrastructure. By varying distance, connectivity and disconnectivity of nodes can be controlled. So, routing is a very important issue in adhoc networks. Each node in the network must be able to take care of routing the data and can discover multihop paths. Routing protocols can be categorized based on methodologies such as (i) Routing information update (ii) Temporal information for routing (iii) Routing topology (iv) Utilization of specific resources.

1.1.1 Classification of Routing Protocols

Routing protocols can be divided into three types as follows:

1) Proactive Routing Protocol
2) Reactive Routing Protocol
3) Hybrid Routing Protocol

Proactive Routing Protocol

Table-driven routing protocols (proactive routing) periodically advertise to all nodes for maintaining up-to-date review of the network. Each node maintains information of other nodes in the routing tables and regularly update information when a node moves. So, these protocols are not suitable for large networks.

Reactive Routing Protocol

On-demand routing protocols (reactive routing) only discovers a new route when it is required. Each node maintains information about other nodes in the routing tables and information is not regularly updated when a node moves. Reactive routing protocols are not suitable for large networks.

Hybrid Routing Protocol

Hybrid routing protocols maintain an efficient balance between both categories of protocols. Nodes within a certain distance from the node concerned, or within a particular geographical region, are said to be within the routing zone of the given node. For routing within this zone, a table driven approach is used. Whereas nodes that are located beyond this zone, an on-demand approach is used. Table-driven schemes are more expensive in terms of energy consumption as compared to the on-demand schemes because of the large routing overhead incurred by the former. Consequently the on-demand approach is preferable for designing minimum energy routing protocols. Fig.1.2 presents the detailed classification of routing protocols in mobile adhoc network.

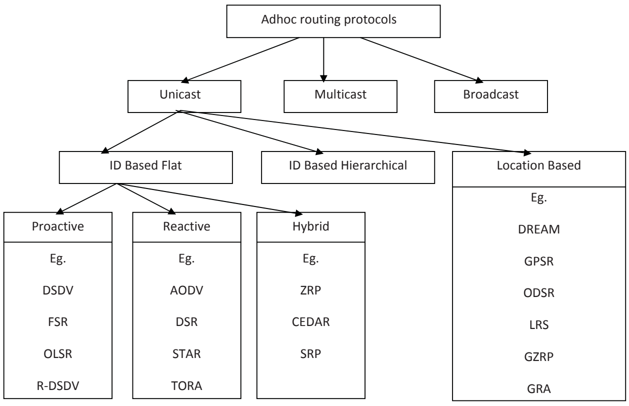


Fig. 1.2. Classifications of Routing protocols

1.2 PRIVACY PRESERVING ROUTING PROTOCOL

With regard to privacy-related notions in communication networks, the terminologies on anonymity, unlinkability, and unobservability are defined here. These notions are defined with regard to item of interest (IOI, including senders, receivers, messages, etc.) as follows:

- **Anonymity** is the state of being not identifiable within a set of subjects, the anonymity set.

- **Unlinkability** of two or more IOIs means these IOIs are no more or no less related with the attacker's view.

- **Unobservability** of an IOI is the state that whether it exists or not is indistinguishable to all unrelated subjects, and subjects related to this IOI are anonymous to all other related subjects.

Anonymity

Existing anonymous routing protocols mainly consider anonymity and partial unlinkability in MANET, most of them exploit asymmetric features of public key cryptosystems to achieve their goals. Complete unlinkability and unobservability are not guaranteed due to incomplete content protection. Existing schemes fail to protect all content of packets from attackers, so that the attacker can obtain information like packet type and sequence number etc. This information can be used to relate two packets, which break unlinkability and may lead to source trace back attacks . Meanwhile, unprotected packet type and sequence number also make existing schemes observable to the adversary.

Unlinkability

Unlinkability alone is not enough in hostile environments like battlefields as important information like packet type is still available to attackers. Then a passive attacker can mount traffic analysis based on packet type. In this case, it is preferable to make the traffic content completely unobservable to outside attackers so that a passive attacker only overhears some random noises. Conversely, this is far from an easy task because it is extremely difficult to hide information on packet type and node identity. Furthermore, a hint on using which key for decryption should be provided with each encrypted packet, which demands careful design to remove linkability. Another drawback of most previous schemes is that relies heavily on public key cryptography, and thus incurs a very high computation overhead.

Unobservability

Among these requirements unobservability is the strongest one in that it implies not only anonymous but also unlinkability. To achieve unobservability, a routing scheme should provide unobservability for both content and traffic pattern.

1.3 TYPES OF UNOBSERVABILITY

Unobservability is classified into two types:

1) Content Unobservability
2) Traffic Pattern Unobservability

1) Content Unobservability – Defines that no useful information can be extracted from the content of any message.

2) Traffic Pattern Unobservability – States that no useful information can be obtained from frequency, length, and source-destination patterns of message traffic. The work will focus on content unobservability, which is orthogonal to traffic pattern unobservability, and it can be combined with mechanisms offering traffic pattern unobservability to achieve truly unobservable communication. The major mechanism to achieve traffic pattern unobservability is traffic padding.

1.4 PHASES OF USOR

USOR propose an efficient privacy-preserving routing protocol USOR that achieves content unobservability by employing anonymous key establishment based on group signature. The setup of USOR is simple. Each node only has to obtain a group signature signing key and an ID-based private key from an offline key server or by a key management scheme . The unobservable routing protocol is then executed in two phases.

1) Anonymous Key Establishment Process
2) Unobservable Route Discovery Process

First, an anonymous key establishment process is performed to construct secret session keys. Then an unobservable route discovery process is executed to find a route to the destination.

USOR is to protect all parts of a packet's content, and it is independent of solutions on traffic pattern unobservability. It can be used with appropriate traffic padding schemes to achieve truly communication unobservability.

1.4.1 Anonymous Key Establishment Scheme

Every node in the adhoc network communicates with its direct neighbors within its radio range for anonymous key establishment. There is a node *S* with a private signing key gskS and a private ID-based key *KS* in the adhoc network and it is surrounded by a number of neighbours within its power range. Following the anonymous key establishment procedure, *S* does the following:

1. $S \rightarrow * : r_sP, SIG_{gsks}(r_sP)$
2. $X \rightarrow S: r_xP, SIG_{gskx}(r_xP), E_{ksx}(k_{x*})$
3. $S \rightarrow X : E_{ksx}(K_{s*})$

(1) *S* generates a random number r_S € Z * q and computes r_SP, where *P* is the generator of G1. It then computes a signature of r_SP using its private signing key gsk_S to obtain SIGgsk_S (r_SP). Anyone can verify this signature using the group public key gpk. It broadcast r_SP, SIGgsk_S (r_SP) within its neighborhood.

(2) A neighbor *X* of *S* receives the message from *S* and verifies the signature in that message. If the verification is successful, *X* chooses a random number r_X € Z * q and computes r_XP. *X* also computes a signature SIGgsk_X ($r_SP|r_XP$) using its own signing key gsk_X. *X* computes the session key $k_{SX} = H_2(r_Sr_XP)$, and replies to *S* with message r_XP, SIGgsk_X ($r_SP|r_XP$),Ek_{SX} ($k_{X*}|r_SP|r_XP$),where k_{X*} is *X*'s local broadcast key.

(3) Upon receiving the reply from *X*, *S* verifies the signature inside the message. If the signature is valid, *S* proceeds to compute the session key between *X* and itself as $k_{SX} = H_2(r_Sr_XP)$. *S* also generates a local broadcast key k_{S*}, and sends Ek_{SX} ($k_{S*}|k_{X*}|r_SP|r_XP$) to its neighbor *X* to inform *X* about the established local broadcast key.

(4) *X* receives the message from *S* and computes the same session key as kSX = H2(rSrXP). It then decrypts the message to get the local broadcast key k_{S*}.The messages exchanged in this phase are not unobservable, but this would not leak any private information like node identities. As a result of this phase, a

pairwise session key k_{SX} is constructed anonymously, which means the two nodes establish this key without knowing who the other party is. Meanwhile, node *S* establishes a local broadcast key k_{S*}, and transmits it to all its neighbors. It is used for per-hop protection for subsequent route discovery.

1.4.2 Privacy Preserving Route Discovery

Privacy-preserving route discovery process based on the keys established in the previous phase. The route discovery process comprises of route request and route reply. The route request messages flood throughout the whole network, while the route reply messages are sent backward to the source node only. There is a node *S* intends to find a route to a node *D*, and *S* know the identity of the destination node *D*.

1.4.2.1 Route Request (RREQ)

S chooses a random number r_S, and uses the identity of node *D* to encrypt a trapdoor information that only can be opened with *D*'s private ID based key, which yields $E_D(S,D, r_SP)$.*S* broadcast the following unobservable route request to its neighbors:

$$Nonce_S, Nym_S, E^{-}k_{S*}(RREQ, N_S, ED(S,D, r_SP), seqno). \qquad (1.1)$$

At the end, *A* prepares and broadcast the following message to all its neighbors:

$$Nonce_A, Nym_A, E^{-}k_{A*} (RREQ, N_A, ED(S,D, r_SP), seqno). \qquad (1.2)$$

Other intermediate nodes do the same as *A* does. Finally,the destination node *D* receives the following message from *C*:

$$Nonce_C, Nym_C, E^{-}k_{C*}(RREQ, N_C, E_D(S,D, r_SP), seqno). \qquad (1.3)$$

1.4.2.2 Route Reply (RREP)

After node *D* finds out the destination node, it starts to prepare a reply message to the source node. For route reply messages, unicast instead of broadcast

are used to save communication cost. Each node also maintains a temporary entry in the routing table.

1.4.2.3 Unobservable Data Packet Transmission

After the source node *S* successfully finds out a route to the destination node *D*, *S* can start unobservable data transmission under the protection of pseudonyms and keys. The data packets from *S* must traverse *A*, *B*, and *C* to reach *D*.

1.5 DESIGN CHALLENGES OF MANET

1. Scalability- the network must preserve its stability. Introducing more nodes in the network means that additional communication messages will be exchanged, so that these nodes are integrated into the existing network.

2. Security- In an attempt to enhance security in MANETs many researchers have suggested and implemented new improvements to the protocols and some of them have suggested new protocols.

3. Routing-finding a feasible path to a destination based on hop length, minimum power required, and lifetime of the wireless link.

4. Multicasting-It plays an important role in emergency search and rescue operations and military communication.

5. Infrastructure- Adhoc network is infrastructureless in which nodes can communicate directly with base station.

6. QoS provisioning- It often requires negotiation between the host and the network, resource reservation schemes, priority scheduling and call admission control.

7. Medium Access Scheme- MAC protocol is to distribute the arbitration for the shared channel for transmission of packets.

8. Power Consumption- Nodes are dependent on battery for their power. Hence power conservation and power management is an important issue in wireless adhoc network.

9. Pricing scheme- As the name suggests a pricing scheme, the adhoc network has a large no of nodes deployed. So if a single node will be very high then the cost of overall network will be very high.

10. Self-Organization-The wireless network is required to perform for self-organization is neighbor discovery, topology organization and topology reorganization.

11. Limited computational power and memory size- It is another factor that affects MANET in the sense that each node store the data individually and sometimes more than one node stored same data and transferred to the base station which waste the power and storing capacity of nodes so we must develop effective routing schemes and protocols to minimize the redundancy in the network.

1.6 LITERATURE SURVEY

In this chapter, papers related to designing unobservable routing protocol for mobile adhoc network and wormhole attack detection using packet leashes, topological detection and statistical analysis are discussed. From that, statistical technique is focused and the homomorphic technique is used to detect the wormhole attack are studied.

1.6.1 ANODR: ANONYMOUS ON DEMAND ROUTING WITH UNTRACEABLE ROUTES FOR MOBILE ADHOC NETWORKS

J.Kong and X.Hong (2003) proposed that, allowing adversaries to trace network routes and nodes at the end of those routes may pose serious threats to the success of hidden missions. Consider for example a battlefield scenario with adhoc, multi-hop wireless communications support. Suppose a covert mission is launched, which includes swarms of reconnaissance, surveillance, and attack task forces. The adhoc network must provide routes between command post and swarms as well as routes between swarms. Providing anonymity and location privacy supports for the task forces is critical, else the entire mission may be compromised [3]. This poses challenging constraints on routing and data forwarding. The adversary could deploy reconnaissance and surveillance forces in the battlefield and maintains communications among them.

On-demand routing schemes are more covert in nature, in that the routing schemes do not advertise in advance. Routing schemes are set up when routes are needed. Nevertheless, the enemy may gain lot of information about the mission by analyzing on-demand routing information and observing packet flows once the connection is established. Since a necessary byproduct of any mission, whether covert or not, is communications across swarms and to/from command

post, these flows and the routes temporarily set up at intermediate nodes must be protected from inference and intrusion.

The purpose of the work is to develop untraceable routes or packet flows in an on-demand routing environment. This goal is very different from other related routing security problems such as resistance to route disruption or prevention of denial-of-service attacks. In fact, the enemy will avoid such aggressive schemes, in the attempt to be as invisible as possible, until it traces, locates, and then physically destroys the assets. The author addresses the untraceable routing problem by a route pseudonymity approach. In this design, the anonymous route discovery process establishes an on-demand route between a source and its destination. Each hop route is associated with a random route pseudonym. Data forwarding in the network is based on route pseudonyms with negligible overhead, local senders and receivers need not reveal their identities in wireless transmission. In other words, the route pseudonymity approach allows us to unlink network member's location and identity. For each route, unlinkability is also ensuring among its route pseudonyms. As a result, in each locality eavesdroppers or any bystander other than the forwarding node can only detect the transmission of wireless packets stamped with random route pseudonyms. It is hard for them to trace how many nodes in the locality, who is the transmitter or receiver, where a packet flow comes from and where it goes to, stay away from the source sender and the destination receiver of the flow.

The design of route pseudonymity is based on a network security concept called broadcast with trapdoor information, which is newly proposed in this work. Trapdoor information is a security concept that has been widely used in encryption and authentication schemes. ANODR is realized upon a hybrid form of these two concepts. The contribution of this work is to present untraceable and intrusion tolerant routing protocol for mobile adhoc networks.

- **Untraceability**

ANODR dissociates adhoc routing from the design of network member's identity/pseudonym. The enemy can neither link network member's identities with their locations, nor follow a packet flow to its source and destination. Though the adversaries may detect the existence of local wireless transmissions, it is hard for them to infer a covert mission's number of participants, as well as the transmission pattern and motion pattern of these participants.

- **Intrusion tolerance**

ANODR ensures there is no single point of compromise in adhoc routing. Node intrusion does not compromise location privacy of other legitimate members, and an on-demand ANODR route is traceable only if all forwarding nodes en route are intruding.

1.6.2 SYBILGUARD: DEFENDING AGAINST SYBIL ATTACKS VIA SOCIAL NETWORKS

Y.Zhu et al (2004) described that Sybil attacks refer to individual malicious users creating multiple fake identities in open-access distributed systems such as peer-to-peer systems. These open-access systems aim to provide service to any user who wants to use the service. When a malicious user's Sybil node comprise a large fraction of the nodes in the system, that one user is able to outvote the honest users in a wide variety of collaborative tasks.

Sybil Attack

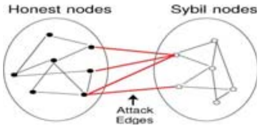


Fig.1.3. Sybil Attack

Sybil attack is shown in the above Fig.1.3.Sybil attacks can be thwarted by a trusted central authority if the authority can tie identities to actual human beings, but implementing such a capability can be difficult or impossible, especially given the privacy concern of the users.

Various Approaches for Detecting Sybil Attack

1) Central authority to impose a monetary charge on each identity.

2) Bind identities to IP addresses or IP prefixes.

3) Require every identity to solve puzzles

First approach is for the central authority to impose a monetary charge on each identity is undesirable in many applications. Without these trusted central authorities, defending against sybil attacks is much harder. Among the small number of approaches, the simplest one perhaps is to bind identities to IP addresses or IP prefixes. Another approach is to require every identity to solve puzzles that require human effort, such as CAPTCHAs. Both approaches can provide only limited protection the adversary can readily steal IP addresses with different prefixes in today's Internet, while CAPTCHAs can be reposted on an adversary's Web site to be solved by users seeking access to that site.

1.6.3 SELF-ORGANIZED PUBLIC-KEY MANAGEMENT FOR MOBILE ADHOC NETWORKS

S.Capkun (2003) presented that mobile adhoc network does not rely on any fixed infrastructure. Instead, all networking functions like routing, mobility management are performed by the nodes themselves in a self-organizing manner. For this reason, securing mobile adhoc networks is challenging task. In this view, there are two extreme ways to introduce security in mobile adhoc networks:

(1) Through a single authority domain, where certificates and/or keys are issued by a single authority, typically in the system setup phase.

(2) Through full self-organization, where security does not rely on any trusted authority or fixed server, not even in the initialization phase.

The effort of this work proposed a self-organizing public-key management system that allows users to create, store, distribute, and revoke their public keys without the help of any trusted authority or fixed server [6]. This approach is developed mainly for open networks, in which users can join and leave the network without any centralized control.

Advantages

Unlike in PGP, where certificates are mainly stored in centralized certificate repositories, certificates in this system are stored and distributed by the nodes in a fully self-organized manner. Each certificate is issued with a limited validity period and therefore contains its issuing and expiration times. Before a certificate expires, its issuer issues an updated version of the same certificate, which contains an extended expiration time. This is called as certificate update. Each node

periodically issues certificate updates, as long as its owner considers that the user-key bindings contained in these certificates are correct.

Drawbacks

The main problem of any public-key based security system is to make each user's public key available to others in such a way that its authenticity is verifiable. In mobile adhoc networks, this problem becomes even more difficult to solve because of the absence of centralized services and possible network partitions. More precisely, two users willing to authenticate each other are likely to have access only to a subset of nodes of the network. The best known approach to the public-key management problem is based on public-key certificates. A public-key certificate is a data structure in which a public key is bound to an identity by the digital signature of the issuer of the certificate.In this system (PGP) user's public and private keys are created by the users themselves. Hence use the same identifier for the user and her node.

Key Authentication

In this system, key authentication is performed via chains of public-key certificates in the following way. When a user *u* wants to obtain the public key of another user *v*, she acquires a chain of valid public-key certificates such that:

1) The first certificate of the chain can be directly verified by *u*, by using a public key that *u* holds and trusts (e.g. own public key).

2) Each remaining certificate can be verified using the public key contained in the previous certificate of the chain.

3) The last certificate contains the public key of the target user *v*.

Certificate Repository

To correctly perform authentication via a certificate chain, a node needs to check that:

(i) all the certificates on the chain are valid

(ii) all the certificates on the chain are correct

(iii) all the certificates contain correct user-key bindings

Types of Certificate Repositories

To find appropriate certificate chains to other users, each node maintains two local certificate repositories:

- The non-updated certificate repository
- The updated certificate repository

The Non-Updated Certificate Repository

The non-updated certificate repository of a node contains expired certificates that the node does not keep updated. The reason for collecting and not updating expired certificates is that most of the certificates will permanently be renewed by their issuers, and only a few will be revoked. Therefore, the non-updated repositories provide the nodes with a very good estimate of the certificate graph. This information helps node to perform authentication.

The Updated Certificate Repository

The updated certificate repository of a node contains a subset of certificates that the node keeps updated. This means that the node requests the updates for the certificates contained in its updated repository from their issuers, when or before they expire. The selection of certificates into the node's updated repository is performed according to an appropriate algorithm.

Certificate Revocation

Certificate revocation is an important mechanism in this scheme. It enables two types of certificate revocation: explicit and implicit. The issuer explicitly revokes a certificate by issuing a revocation statement and by sending it to the nodes who stored the certificate in question. The implicit revocation relies on the expiration time contained in the certificates. Every certificate whose expiration time passes is implicitly revoked. This second mechanism is straightforward, but requires some loose time synchronization of the nodes.

1.6.4 ANONYMOUS SECURE ROUTING IN MANET

B.Zhu (2004) described that, Anonymous Secure Routing (ASR) protocol that cannot only protect the privacy of nodes and routes, but also ensure the security of discovered routes. Detailed analysis is given to show that ASR can ensure anonymity and security of the routing protocol against known passive and active attacks.

Merits

Stronger location privacy.

Demerits

Make use of one time public/private key pairs to achieve anonymity and unlinkability.

1.6.5 WORMHOLE ATTACK

F.Nait-Abdesselam (2008) proposed that, a wormhole attack is a particularly severe attack on MANET routing where two attackers, connected by a high-speed off-channel link, are strategically placed at different ends of a network. These attackers then record the wireless data they overhear, forward it to each other, and replay the packets at the other end of the network [2]. Replaying valid network messages at improper places, wormhole attackers can make far apart nodes believe they are immediate neighbors, and force all communications between affected nodes to go through them.

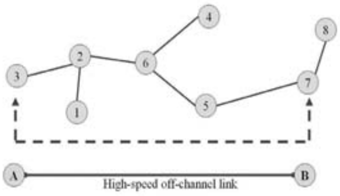


Fig.1.4. Wormhole attack

In Fig.1.4, when node 3 sends a HELLO message, intruder A forwards it to the other end of the network, and node 7 hears this HELLO message. Since 7 can hear a HELLO message from 3, it assumes itself and node 3 to be direct neighbors. Thus, if 7 want to forward anything to 3, it will do so through the wormhole link effectively giving the wormhole attackers full control of the communication link.

In Fig.1.4, if 3 want to communicate with 7, it sends out a request, which a wormhole, once again, forwards without change to the other end of the network, directly to node 7. A request also travels along the network in a proper way, so 7 is lead to believe it has two possible routes to node 3. A 4-hop route through nodes 2, 6, and 5, and a single-hop direct link. Protocols will then select the shortest route, once again giving wormhole attackers full control of the link.

Majority of adhoc routing protocols rely on the correctness of their neighbors information for routing decisions, thus allowing wormhole-induced disruptions to have greater effects. For example, in the situation described in Fig. 1.4, where nodes 3 and 7 think they are direct neighbours, nodes 5 and 8 will then think they are two hops away from node 3 (going through node 7), and will communicate with node 3 through the wormhole link.

Wormhole Attack Modes

Wormhole attacks can be launched using several modes.

- Wormhole using Encapsulation
- Wormhole using Out-of-Band Channel

Wormhole Using Encapsulation

Wormhole using encapsulation technique is shown in Fig.1.5.When the source node broadcast the RREQ packet, a malicious node which is at one part of the network receives the RREQ packet. It tunnels the packet to a second colluding party which is at a distant location near the destination, it then rebroadcasts the RREQ. The neighbors of the second colluding party receive the RREQ and drop any further legitimate requests that may arrive later on legitimate multihop paths.

The result is that the routes between the source and the destination go through the two colluding nodes that will be said to have formed a wormhole between them. It prevents nodes from discovering legitimate paths that are more than two hops away.

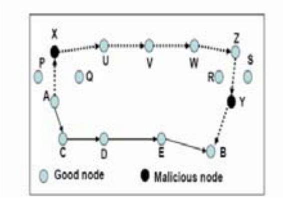


Fig.1.5. Wormhole Using Encapsulation

Wormhole Using Out-Of-Band Channel

Wormhole attack involves the use of an out of band channel here.Fig.1.6 shows the wormhole attack using out of band channel. This attack is launched by having an out-of-band high bandwidth channel between the malicious nodes. This mode of attack needs specialized hardware capability.

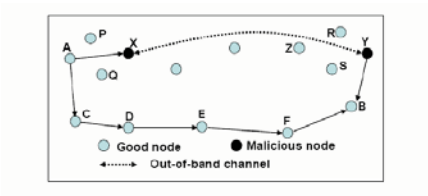


Fig.1.6. Wormhole through out-of band channel

1.6.6 WORMHOLE DETECTION USING PACKET LEASHES

Packet Leash

Y.Hu (2003) described that, Leash is any information that is added to the packet designed to restrict the packets maximum allowed transmission distance. Two types of leashes used.

- Geographic Leashes
- Temporal Leashes

Geographic Leashes

Geographic Leashes ensures that the recipient of the packet is within a certain distance from the sender. Each node must know its location and all the nodes must have loosely synchronized clocks. Sending node must include its own location ($p_{s)}$ and time at which it sends the packet (t_s) in the packet. Receiving node compares these values to its own location p_r and time $t_{r.}$

Constraints

Clocks must be synchronized within ±Δ.An upper bound velocity is calculated. Distance between sender and receiver is calculated based on t_s and t_r.Diginal signature is used for authentication.

Requirements

Requires GPS co-ordinates of every node and loosely synchronized clocks. The main advantage is it is a straightforward solution and it is robust. Disadvantage is it inherits the general limitations of GPS technology.

Temporal Leashes

It ensures that the packet has an upper bound on its lifetime, which restricts the maximum travel distance. It implements the packet expiration time. Sender sends the packet at local time t_s it needs to set the packet expiration time to t_e=ts+L/c-Δ.TIK-TESLA with instant key disclosure allows the sender to disclose the key in the same packet. It requires time synchronization. Symmetric cryptographic primitive is used.

Modules

- Sender Set up
- Receiver Bootstrapping
- Sending and Verifying authenticated packet

Requirements

Temporal Leashes requires tightly synchronized clocks and the main disadvantage is time synchronization level is not achievable.

1.6.7 TOPOLOGICAL DETECTION

N.S.Raote (2011) analyzes the wormhole problem using a topology methodology and proposes an effective distributed approach which relies on network connectivity information without any requirements on special hardware device or any rigorous assumptions on network properties.

Detection Algorithm

1) Construct a shortest path tree from an arbitrarily selected root node.Then only each node obtains shortest path to the root.
2) Select the candidate loops from the cut pair on the shortest path tree.
3) Detect and locate class I and II worms by testing whether a candidate loop is an independent non separating loop.
4) Check whether there exists a cycle that crosses the loop at one time or not.
5) Check the existence of worms by seeking the non-separating loop pairs.

Design Challenges

- Candidate loop selection
- Finding independent non-separating loops
- Seeking knit non-separating loop pairs.

1.6.8 DETECTION BASED ON HARDWARE DEVICES

Time of Flight

Requires Hardware enabling one-bit message and immediate replies without CPU involvement.

Demerit

It requires MAC layer modifications.

Directional Antennas

Directional antennas on all nodes or several nodes with GPS.

Demerit

Good solutions for networks relying on directional antennas, but not directly applicable to other networks.

Network Visualization

Network visualization is not readily applicable to mobile networks. Network visualization requires centralized controller.

Demerit

- Seems promising
- Works best on dense networks
- Mobility not studied
- Varied terrains not studied

Detection Based on Localization

Localization concepts worked on the basis of distance vector based routing protocol. The wormhole attack detection using distance vector based routing protocol is shown in the Fig.1.7.

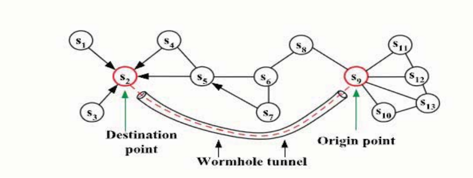


Fig. 1.7.Wormhole against distance vector based routing protocol

Requirements

- Location aware guard nodes
- Closest Guard

1.6.9 DETECTION BASED ON STATISTICAL ANALYSIS

Qian.L (2005) described that the statistical analysis works only with multipath on demand routing protocols.

Various Techniques of Statistical Analysis

- Link frequency analysis
- Trust based model
- Homomorphic encryption

Link Frequency Analysis

In link frequency analysis, an abnormal high frequency of the link suggests a wormhole presence.

Trust Based Model

- Broadcast Route Request
- Append trust vectors
- Send Route Reply
- Check for suspicious link

Homomorphic Encryption

Homomorphic encryption is a form of encryption where a specific algebraic operation performed on the plaintext is equivalent to another (possibly different) algebraic operation performed on the ciphertext. The homomorphic property of various cryptosystems can be used to create secure voting systems, collision-resistant hash functions, private information retrieval schemes and enable widespread use of cloud computing by ensuring the confidentiality of processed data.

CHAPTER 2

IMPLEMENTATION

2.1 PROBLEM DESCRIPTION

Privacy protection of mobile ad hoc networks is more demanding than that of wired networks due to the open nature and mobility of wireless media. In wired networks, one has to gain access to wired cables so as to eavesdrop communications. In contrast, the attacker only needs an appropriate transceiver to receive the wireless signal without being detected. In wired networks, devices like desktops are always static and do not move from one place to another. Hence in wired networks there is no need to protect users' mobility behavior or movement pattern, while this sensitive information should be kept private from adversaries in wireless environments. Otherwise, an adversary is able to profile users according to their behaviors, and endanger or harm users based on such information. Lastly, providing privacy protection for ad hoc networks with low-power wireless devices and low-bandwidth network connection is a very challenging task.

Mobile ad hoc networks are comprised of nodes that must cooperate to dynamically establish routes using wireless links. Routes may involve multiple hops with each node acting as a host and router. Since ad hoc networks typically work in an open un-trusted environment with little physical security, they are subject to a number of unique security attacks like wormhole attack. The wormhole attack is considered to be a serious security attack in multi-hop ad hoc networks. In wormhole attack, attacker makes tunnel from one end of the network to the other, nodes stay in different location on two ends of tunnel believe that they are true

neighbors and makes conversation through the wormhole link. Unlike many other attacks on ad-hoc routing, a wormhole attack cannot be prevented with cryptographic solutions because intruders neither generate new, nor modify existing, packets, but rather forward existing ones. The proposed technique makes use of variance in routing information between neighbors to detect wormholes. The base of dissertation is to record all the possible paths from source to destination and check if any new route is updated by the node. Then it will, calculate the shortest path and isolate the node which establishes the new route as wormhole nodes.

2.2 OVERVIEW OF THE PROJECT

The basic idea of the technique is to discover alternative routes to a target node T that is one-hop neighbor's nodes that do not go through the wormhole. These alternative routes will be extensively dissimilar in length, means the length of the alternative path is greater than the path that have wormhole, and otherwise the wormhole will not attract large amounts of traffic. Consider a communicating source-destination node pair (S, D) as in Fig. 2.1, with source route P(S, D). If node S want to detect the existence of a wormhole, S would find out a new route to D and if the length of the new route differs extensively compared to the length of P(S,D), it is concluded that wormhole exists.

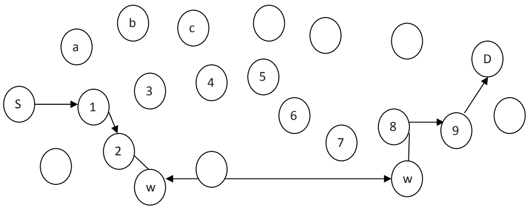


Fig. 2.1. Malicious Route Containing Wormhole

Consider a situation in which the source route found by any routing protocol is S->1->2->3->4->5->6->7->8->9->D as shown in Fig.2.1.This is the legal path. With the introduction of closed wormhole attack in the legal source route, the new malicious route will be S->1->2->8->9->D.

As the proposed approach for detection of wormhole solely depends on the density of nodes in the network, therefore the value of threshold must be predefined. The threshold value is calculated by checking the average number of hops between the nodes in the network. In the situation defined in Fig.2.1, the value of threshold is calculated as X.

Now in order to detect the wormhole, the proposed approach attempts to find the number of hops on the second shortest route between two alternate nodes starting from the source S. If number of hops in the second shortest path is greater that the predefined threshold, then it is declared that the wormhole is present between the two nodes.

For example the malicious path containing the closed wormhole is S->1->2->8->9->D. Now proposed algorithm will check the number of hops found in the second shortest path between the nodes S and 2. In the situation given in Fig.2.1, second shortest path between S and 2 will S->a->b->c->2. Now the number of hops between S and 2 is found to be 3 which is less than predefined threshold (i.e. X), it is declared that no wormhole is present between the nodes S and 2.

The same algorithm is followed by next two nodes (i.e. 1 and 8). As the number of hops in the second shortest path from node 1 to 8 is greater than the predefined threshold, therefore it the presence of wormhole is declared between the nodes 1 and 8.

A novel approach for detecting wormhole attacks is purely based on local connectivity information. Such information is often collected any way by various upper layer protocols such as routing, thus may not present any additional overhead. No additional hardware object is needed making the approach universally applicable. No timing analysis is done ensuring that even physical layer attacks can detected. A proposed technique does not use location information and is able to detect attacks that are launched even before the network is set up, that may influence localization. The following steps are focused here.

1. Initially the topology is constructed based on node information.

2. After the topology is created, the nodes are connected using the given connectivity model.

3. Once the connectivity graph is established, the following experiments are performed.

a) Connectivity in the entire network is checked. The network is assumed disconnected if any two nodes do not have a path to each other.

b) The wormhole detection algorithm is run to see whether there is any false route information is added newly. (At this time, the technique is independent of whether the entire network is connected or not, connected networks are more useful from a practical standpoint)

c) A wormhole attack is established between two randomly chosen locations. The algorithm is run again to see whether it detects the wormhole. After the detection of wormhole this algorithm isolates the nodes as wormhole nodes which created new route from the second hop.

2.3 MODULES OF THE PROJECT

2.3.1 Network Configuration

In Network Configuration, the system is register to network topology. System login to the network topology while it checks the user authentication that is specifying system IP address, port number and status. Then only server system, allows the node in to the transmission .To send the packets to the destination a source can multicast a RREQ packets to all the nodes as in Fig. 2.2.After the route established a system can send the packets to the destination. System connection details also maintain in server.

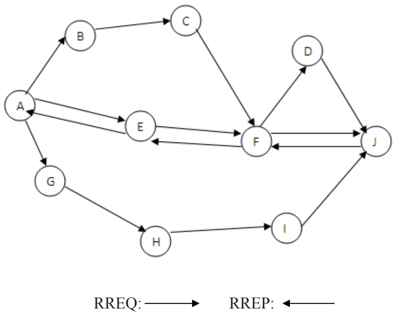


Fig. 2.2. Network Construction

2.3.2 Network Management

In Network Management, server monitors the whole network. System connection process also monitor by server. Server maintains the path details. Suppose any two systems made connection between each other while after system login means, these system details maintain separately in server side. System join and reliving process also monitor by server in network.

2.3.3 Unobservable Routing Scheme

In this module, the available path is calculated for transmission. At that time to avoid the two hop problem, some nodes are avoided on path which is maintained separately in the server, because these nodes made connection after system login. These nodes may be act as hacker in the path that's why these nodes are avoided.

2.3.4 Message Transmission

In Message Transmission, transmit message after calculating best path from available path. Messages pass through the intermediate systems. The best path is selected based on the throughput that means speed of the packet transmission on path. Massage packets are reached receiver side when it can store in memory. Save dialog box opened for store process.

2.4 RC4 ALGORITHM

The RC4 algorithm is remarkably simply and quite easy to explain. A variable-length key of from 1 to 256 bytes (8 to 2048 bits) is used to initialize a 256-byte state vector S, with elements S[0], S[1], …, S[255]. At all times, S contains a permutation of all 8-bit numbers from 0 through 255. For encryption and decryption, a byte *k* is generated from S by selecting one of the 255 entries in a systematic fashion. As each value of *k* is generated, the entries in S are once again permuted.

Initialization of S

To begin, the entries of S are set equal to the values from 0 through 255 in ascending order. That is S[0] = 0, S[1] = 1, …, S[255] = 255. A temporary vector, T, is also created. If the length of the key K is 256 bytes, then K is transferred to T. Otherwise, for a key of length keylength bytes, the first key length elements of T are copied from K and then K is repeated as many times as necessary to fill out T. These preliminary operations can be summarized as follows:

```
/* Initialization */
for i = 0 to 255 do
S[i] = i;
T[i] = K[i mod keylen];
```

T to produce the initial permutation of S. This involves starting with S[0] and going through to S[255], and, for each S[i], swapping S[i] with another byte in S according to a scheme dictated by T[i]:

```
/* Initial Permutation of S */
j = 0;
for i = 0 to 255 do
j = (j + S[i] + T[i]) mod 256;
Swap (S[i], S[j]);
```

Because the only operation on S is a swap, the only effect is a permutation. S still contains all the numbers from 0 through 255.

Stream Generation

Once the S vector is initialized, the input key is no longer used. Stream generation involves starting with S[0] and going through to S[255], and, for each S[i], swapping S[i] with another byte in S according to a scheme dictated by the current conFig.uration of S. After S[255] is reached, the process continues, starting over again at S[0]:

```
/* Stream Generation */
i, j = 0;
while (true)
i = (i + 1) mod 256;
j = (j + S[i]) mod 256;
Swap (S[i], S[j]);
t = (S[i] + S[j]) mod 256;
k = S[t];
```

To encrypt, XOR the value *k* with the next byte of plaintext. To decrypt, XOR the value *k* with the next byte of ciphertext.

CHAPTER 3

RESULTS AND DISCUSSION

3.1 SYSTEM SPECIFICATION

3.1.1 Hardware Requirements

Processor	: Pentium III and above
Clock speed	: 550MHz
Hard Disk	: 20GB
RAM	: 128MB or above
Cache Memory	: 512KB
Monitor	: Color Monitor
Keyboard	: 104Keys
Mouse	: 3Buttons

3.1.2 Software Requirements

Operating System	: Windows Family.
Language	: Java
Data Bases	: Microsoft SQL Server
Front End	: Java Swing

3.3 RESULT ANALYSIS

The objective is to determine how many pairs of source and destination nodes do not have paths in a given set of nodes. The network density is important for ad-hoc networks. For a given region, there are two ways to increase the density. (1) Increase the number of nodes or (2) Increase the transmission range of the nodes. Since the complexity of the simulation program depends on the number of nodes, the number of nodes is fixed and different range values are used. The proposed approach is implemented in Java swing on a Pentium-III PC with 20 GB hard-disk and 256 MB RAM. The proposed approach shows efficient results of retrieving data from mobile nodes and has been efficiently tested on different systems.

This implementation has shown that the prevention can be considered as reliable. The routing metric packet delivery is high mobility. Characteristics and results for this system were achieved after an extensive design part in our implementation. Design has been a key part to get reliable results. It could be achieved by splitting data packets from the source to the destination. The whole message would not be transmitted by the same path or the same nodes all the time. Another solution could be to enforce reliability adding some redundancy code. In that case, it would allow not sending again packets in case one link breaks.

3.3.1 WORMHOLE DETECTION RESULT

Wormhole attack is detected by running the algorithm to find an alternative path to the two hop neighbor node. If the hop count of that path is greater than the threshold this path will consider as a wormhole path. Threshold is calculated by checking the average number of hops between the nodes. When threshold value is increased then the detection ratio of wormhole shows good result

which clarify that the proposed technique is detect wither shortest path given by AODV having wormhole or not and it will help network administrator for taking the decision wither selected path is legal or not. From the analysis the overhead of proposed technique is more compared to the existing AODV. Similarly when the number of nodes increases the routing load is also increases.

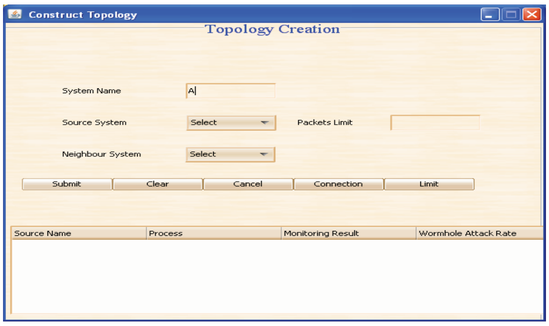
3.3.2 ENCRYPTION RESULT

Table 3.1 Speed Comparison of symmetric ciphers

CIPHER	KEYLENGTH	SPEED(Mbps)
DES	56	9
3DES	168	3
RC2	Variable	0.9
RC4	Variable	45

Table 3.1, compares execution times of RC4 with three well-known symmetric block ciphers. The advantage of a block cipher is that reuse of keys. However, if two plaintexts are encrypted with the same key using a stream cipher, then cryptanalysis is often quite simple.

3.5 SNAPSHOTS



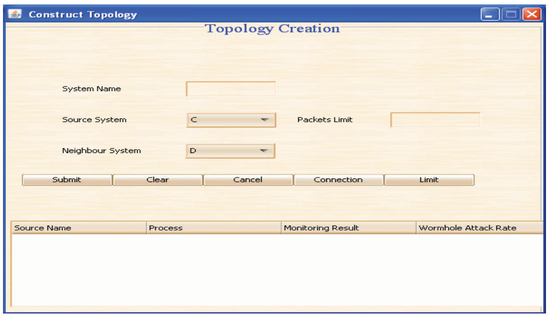
System Name: A

Source System: Select

Neighbour System: Select

Packets Limit

Source Name	Process	Monitoring Result	Wormhole Attack Rate



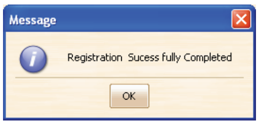
System Name

Source System: C

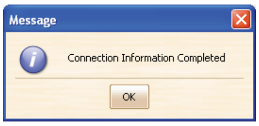
Neighbour System: D

Packets Limit

Source Name	Process	Monitoring Result	Wormhole Attack Rate

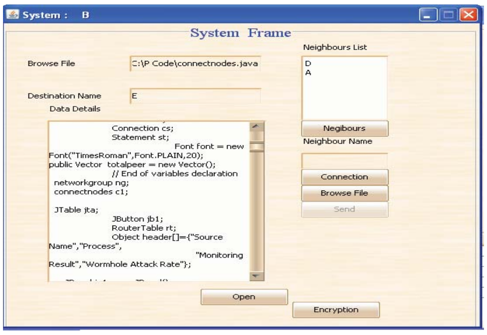


Registration Sucess fully Completed



Connection Information Completed

Fig. 3.1 Network Setup



Browse File: C:\P Code\connectnodes.java

Destination Name: E

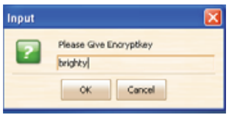
Data Details

```
                    Connection cs;
                    Statement st;
                                        Font font = new
Font("TimesRoman",Font.PLAIN,20);
public Vector  totalpeer = new Vector();
                    // End of variables declaration
  networkgroup ng;
  connectnodes c1;

  JTable jta;
                    JButton jb1;
                    RouterTable rt;
                    Object header[]={"Source
Name","Process",
                                              "Monitoring
Result","Wormhole Attack Rate"};
```

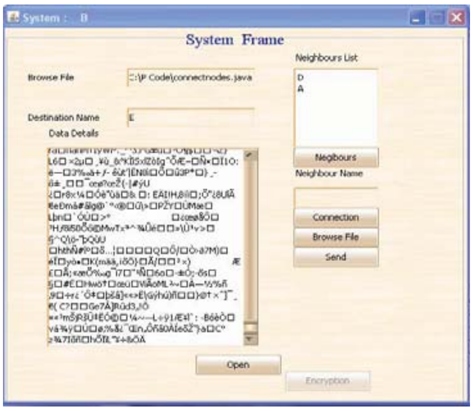
Neighbours List: D, A

Neighbour Name



Please Give Encryptkey

brighty



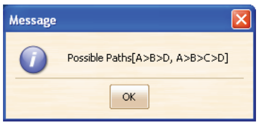
Browse File: C:\P Code\connectnodes.java

Destination Name: E

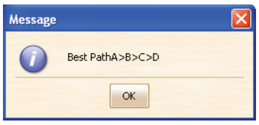
Data Details

Neighbours List: D, A

Neighbour Name

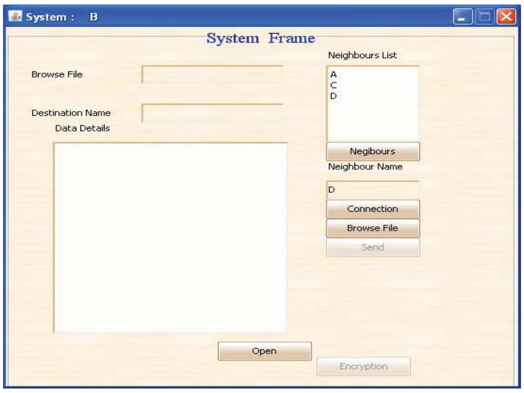


Possible Paths[A>B>D, A>B>C>D]



Best PathA>B>C>D

Fig. 3.2.Path Calculation & Message Encryption



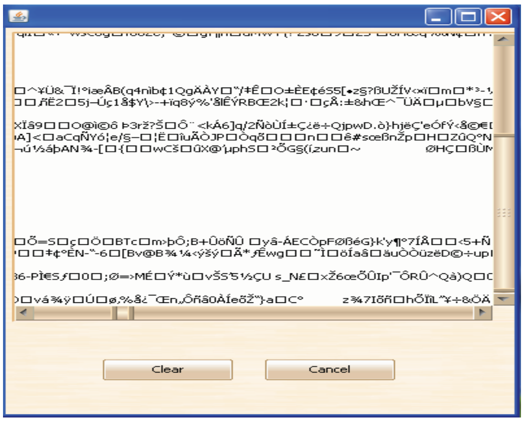
Browse File

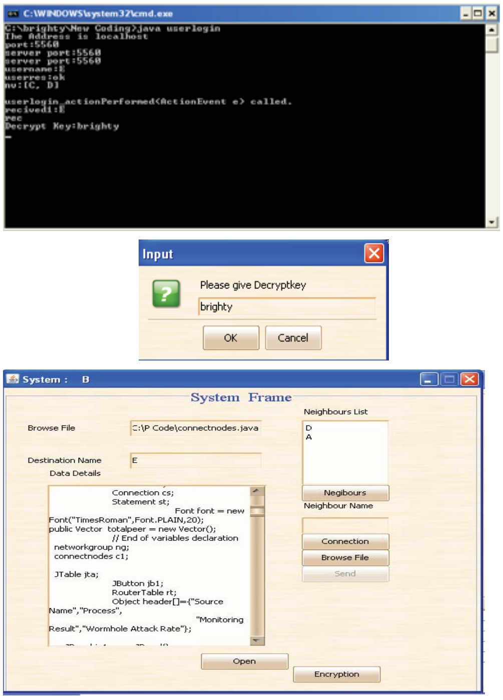
Destination Name

Data Details

Neighbours List: A, C, D

Neighbour Name: D





```
C:\brighty\New Coding>java userlogin
The Address is localhost
port:5560
server port:5560
server port:5560
username:E
userres:ok
nv:[C, D]

userlogin_actionPerformed(ActionEvent e) called.
recived1:E
rec
Decrypt Key:brighty
_
```

Please give Decryptkey: brighty

Browse File: C:\P Code\connectnodes.java

Destination Name: E

Neighbours List: D, A

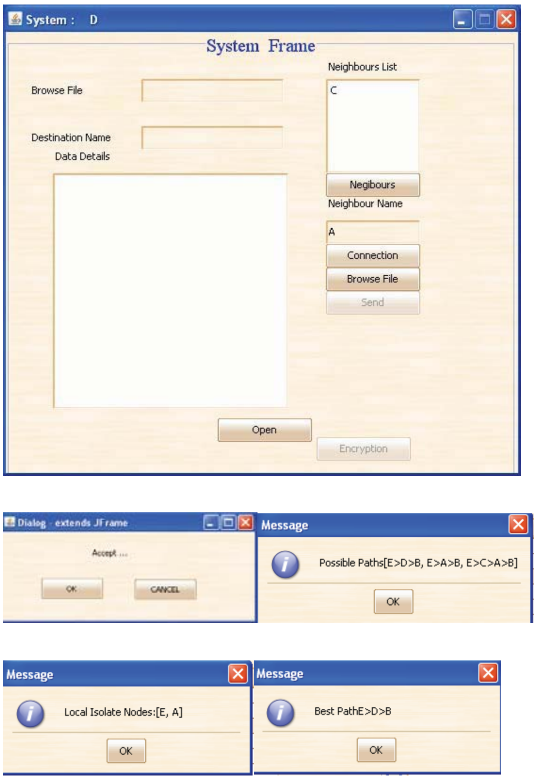
Neighbour Name:

Data Details:

```
                Connection cs;
                Statement st;
                                Font font = new
Font("TimesRoman",Font.PLAIN,20);
public Vector  totalpeer = new Vector();
                // End of variables declaration
 networkgroup ng;
 connectnodes c1;

 JTable jta;
                JButton jb1;
                RouterTable rt;
                Object header[]={"Source
Name","Process",
                                          "Monitoring
Result","Wormhole Attack Rate"};
```

Fig. 3.3 Message Decryption



Browse File:

Destination Name:

Data Details:

Neighbours List: C

Neighbour Name: A

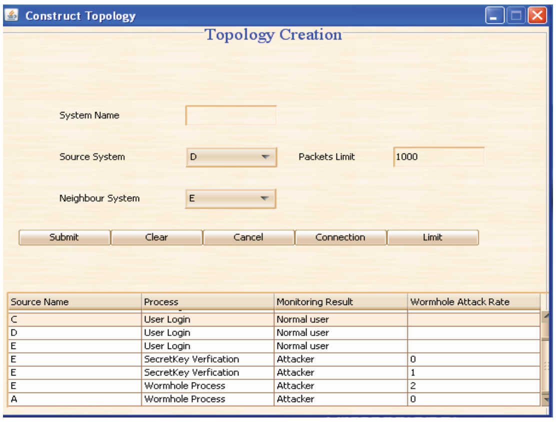
Accept ....

Possible Paths[E>D>B, E>A>B, E>C>A>B]

Local Isolate Nodes:[E, A]

Best PathE>D>B

Fig. 3.4.Wormhole Process



System Name:

Source System: D

Packets Limit: 1000

Neighbour System: E

Source Name	Process	Monitoring Result	Wormhole Attack Rate
C	User Login	Normal user	
D	User Login	Normal user	
E	User Login	Normal user	
E	SecretKey Verfication	Attacker	0
E	SecretKey Verfication	Attacker	1
E	Wormhole Process	Attacker	2
A	Wormhole Process	Attacker	0

Fig. 3.5 Monitoring Process

3.6 CONCLUSION & FUTURE WORK

A simple technique for detecting wormholes in ad hoc network is identified. This method employs routing variation between neighbors to determine the existence of a wormhole. The technique is localized, requires only a small overhead, and does not have special requirements such as location information, accurate synchronization between nodes, special hardware. The proposed approach was implemented in Java. The proposed approach concepts show efficient results of retrieving data from mobile nodes. Thus this approach is very effective in detecting wormhole attack based on the connectivity information. The future work includes developing a technique for removal of the wormhole attack.

APPENDIX

SAMPLE CODE

Connectnodes.java

```
 import java.awt.*;
 import java.awt.event.*;
 import javax.swing.*;
 import java.sql.*;
 import java.util.*;
 import java.net.*;
 import java.io.*;
 import javax.swing.border.*;
 import javax.swing.table.*;
 public class connectnodes extends JFrame
 {
 // Variables declaration
private JLabel jLabel1;
private JLabel jLabel2;
private JLabel jLabel3;
private JLabel jLabel4,jLabel5;
private JTextField jTextField1;
private JComboBox jComboBox1;
private JComboBox jComboBox2;
private JTextField jTextField2;
public JTextField jTextField5;
private JButton jButton1;
private JButton jButton2;
```

```
private JButton jButton3;
private JButton jButton4,jButton5;
private JButton jButton7;
private JButton monitor;
private JButton limit;
public int limitvalue=0;
 private JPanel contentPane;
private JPanel panel;
public Runtime r;
int n;
int i=0;
int portno;
String nodename = "",node,des;
String sysname = "";
ResultSet rs;
Connection cs;
Statement st;
Font font = new Font("TimesRoman",Font.PLAIN,20);
public Vector  totalpeer = new Vector();
// End of variables declaration
 networkgroup ng;
connectnodes c1;
JTable jta;
JButton jb1;
RouterTable rt;
Object header[]={"Source Name","Process","Monitoring Result","WormholeAttack
Rate"};
JPanel jp1=new JPanel();
```

```
JScrollPane jsp;
JLabel pic=new JLabel();
int flag=1;
public connectnodes(networkgroup ng1)
 {
super();
initializeComponent();
ng=ng1;
this.setVisible(true);
}
public void initializeComponent()
{
jLabel1 = new JLabel();
jLabel2 = new JLabel();
jLabel3 = new JLabel();
 jLabel4 = new JLabel();
jLabel5 = new JLabel();
panel=new JPanel();
jTextField1 = new JTextField();
jTextField2 = new JTextField();
jTextField5 = new JTextField();
jButton1 = new JButton();
jButton2 = new JButton();
Button3 = new JButton();
jButton4 = new JButton();
jButton5 = new JButton();
monitor = new JButton();
```

```
 limit = new JButton();
jButton7 = new JButton();
jta=new JTable();
rt=new RouterTable();
 rt.setColumnIdentifiers(header);
 jta.setModel(rt);
jsp=new JScrollPane(jta);
jComboBox1 = new JComboBox();
jComboBox2 = new JComboBox();
contentPane = (JPanel)this.getContentPane();
jComboBox1.addItem("Select");
ComboBox2.addItem("Select");
jLabel2.setText("System Name");
jLabel3.setText("Source System");
jLabel4.setText("Neighbour System");
jLabel5.setText("Packets Limit");
jButton1.setText("Submit");
jButton1.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
jButton1_actionPerformed(e);
}
});
jButton2.setText("Clear");
jButton2.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
```

```
jTextField1.setText("");
jTextField2.setText("");
jComboBox1.setSelectedItem("Select");
jComboBox2.setSelectedItem("Select");
}
});
jButton3.setText("Cancel");
jButton3.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
dispose();
}
});
jButton5.setText("Connection");
jButton5.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
jButton5_actionPerformed(e);
}
});
jButton7.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
try
{}
catch(Exception e1)
{}
```

```
}
});
// contentPane
limit.setText("Limit");
limit.addActionListener(new ActionListener() {
public void actionPerformed(ActionEvent e)
{
try
{
limitvalue=Integer.parseInt(jTextField5.getText());
}
catch(Exception e1)
{
}
}
});
contentPane.setLayout(null);
panel.setLayout(null);
addComponent(contentPane,panel, -5,-5,600,500);
addComponent(panel, jLabel2, 65,100,100,25);
addComponent(panel, jLabel3, 65,150,100,25);
addComponent(panel, jLabel4, 65,200,150,25);
addComponent(panel, jTextField1, 204,100,100,25);
addComponent(panel, jComboBox1, 204,150,100,25);
addComponent(panel,jComboBox2, 203,200,100,25);
addComponent(panel,jLabel5, 325,150,150,25);
addComponent(panel,jTextField5, 430,150,100,25);
```

```
addComponent(panel, jButton1, 23,250,100,19);
addComponent(panel, jButton2, 123,250,100,19);
addComponent(panel, jButton3, 223,250,100,19);
addComponent(panel, limit, 423,250,100,19);
addComponent(panel, jsp, 10,325,600,140);
addComponent(panel, jButton5,323,250,100,19);
Border etched=BorderFactory.createtchedBorder();
Borderborder=BorderFactory.creatTitledBorder(etched,"TopologyCreation",Titled
Border.CENTR,TitledBorder.DEFAULT_JUSTIFICATION,font,Color.blue)
panel.setBorder(border)
this.setTitle(" Construc Topology");
this.setLocation(new oint(100, 100));
this.setSize(new Diension(600, 500));
this.setDefaultCloseperation(DO_NOTHING_ON_CLOSE);
this.setResizablefalse);
}
private void addComponent(Container container,Component c,int x,inty,int
width,int height)
{
c.setBounds(x,y,width,height);
container.add(c);
}
 private void jButton1_actionPerformed(ActionEvent e)
{
nodename = jTextField1.getText()
String cost = ""
if(!nodename.equals(""))
{
try
```

```
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
cs=DriverManager.getConnection("jdbc:odbc:USOR","sa","");
st=cs.createStatement();
rs= st.executeQuery("select * from NodeInformation where NodeName LIKE
'"+nodename+"'" );  //OR SystemName LIKE '"+sysname+"'
 if(rs.next()) {
OptionPane.showMessageDialog(this,"The given Data already Exists");
i--;
 }
 else
{        String query = "insert into NodeInformation
values('"+nodename+"','0','','OFF','','"+cost+"')";
System.out.println("cost:"+cost);
st.execute(query);
JOptionPane.showMessageDialog(this,"Registration  Sucess fully Completed");
jComboBox1.addItem(nodename);
jComboBox2.addItem(nodename);
jTextField1.setText("");
jTextField2.setText("");
  }
}
catch(Exception ee)
{
JOptionPane.showMessageDialog(this,"Specify the Correct PortNo");
System.out.println("Connectivity Error");
ee.printStackTrace();
i--;
```

```
}
}
else
{
 JOptionPane.showMessageDialog(this,"Specify correct details");
}
}
private void jButton5_actionPerformed(ActionEvent e)
{
try
{
node = (String)jComboBox1.getSelectedItem();
 des = (String)jComboBox2.getSelectedItem();
System.out.println(""+node);
System.out.println(""+des);
if(node.equals("Select")||des.equals("Select")){
JOptionPane.showMessageDialog(this"Specify the Nodes");
}
else if (node.equals(des))
{
OptionPane.showMessageDialog(this,"Specify a Valid Neighbour");
}
else
{
try
{
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
```

```
cs=DriverManager.getConnection("jdbc:odbc:USOR","sa","");
st=cs.createStatement();
rs= st.executeQuery("select * from Connection where NodeName LIKE '"+node+"'
AND Neighbour LIKE '"+des+"'");  //OR SystemName LIKE '"+sysname+"'
System.out.println("1");
if(rs.next())
JOptionPane.showMessageDialog(this,"The given Data already Exists");
}
else
String query = "insert into Connection values('"+node+"','"+des+"','1','0','OFF')";
st.executeUpdate(query);
query = "insert into Connection values('"+des+"','"+node+"','1','0','OFF')";
st.executeUpdate(query);
JOptionPane.showMessageDialog(this,"Connection InformationCompleted");
}
catch(SQLException ee)
JOptionPane.showMessageDialog(this,"Connectivity Error");
System.out.println("Connectivity Error");
ee.printStackTrace();
}
}
}
catch (Exception e3)
{
JOptionPane.showMessageDialog(this,"Exception");
e3.printStackTrace();
}
}
```

```
public  void adddat(Vector dat)
{
System.out.println("adddat");
rt.addRow(dat);
system.out.println("adddat");
}
class RouterTable extends DefaultTableModel
{
RouterTable()
{
}
}
public Vector connect()
{
  try
 {
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
 cs=DriverManager.getConnection("jdbc:odbc:USOR","sa","");
 st=cs.createStatement();
ResultSet rs=st.executeQuery("select NodeName from NodeInformation");
 totalpeer.add("Select");
while(rs.next())
 {
    totalpeer.addElement(rs.getString(1).trim());
    System.out.println(""+totalpeer);
       }
 }
```

```
   catch (Exception ex)
   {        ex.printStackTrace();
    }
   return totalpeer;
 }}
```

REFERENCES

1. Capkun S., Buttyan L., and Hubaux J., “Self-organized public-key management for mobile adhoc networks,” IEEE Trans. Mobile Comput., vol. 2, no. 1, pp. 52–64, Jan.-Mar. 2003.
2. Hu Y., Perrig A., and Johnson D., “Packet Leashes: A Defense Against Wormhole Attacks in Wireless Adhoc Networks”, in Proc. of INFOCOM 2003, San Francisco, CA, USA, April 2003.
3. Kong.J and Hong.X, “ANODR: aonymous on demand routing with untraceable routes for mobile ad-hoc networks,” in Proc. ACM MOBIHOC’ 03, pp. 291–302.
4. Krishna Rao T., Mayank Sharma, Dr. M. V. Vijaya Saradhi,“Securing Layer-3 Wormhole Attacks in Ad-Hoc Networks “, International Journal of Modern Engineering Research (IJMER) , Vol.2, Issue.1, Jan-Feb 2012 pp-230-234 ISSN: 2249-6645.
5. Nait-Abdesselam.F, Bensaou.B, and Taleb. T, “Detecting and avoiding Wormhole attacks in Wireless Ad-hoc Networks”, in IEEE Communication Magazine.vol.46, April 2008, pp.127-133.
6. Qian. L., Song, N. Li, X. “Detecting and Locating Wormhole Attacks in Wireless Adhoc Networks through Statistical Analysis of Multi-path”. In IEEE WCNC 2005, New Orleans, LA, USA, March 13-17, pp. 2106–2111,2005.
7. Raote N.S., Hande K.N., Seo J., “Approaches towards Mitigating Wormhole Attack in Wireless Adhoc Networks”, International Journal of Advanced Engineering Sciences and Technology (IJAEST) ,vol.no.2,Issue no.2, pp. 172-175, 2011.
8. ZhiguoWan, Kui Ren, Ming Gu “USOR: An Unobservable Secure On-Demand Routing Protocol for Mobile Ad Hoc Networks”, IEEE Transactions On Wireless Communications, vol. 11, no. 5, May 2012.
9. Zhu.B, Wan.Z, Bao.F, Deng.R.H, and KankanHalli.M, “Anonymous secure routing in mobile ad-hoc networks,” in Proc. 2004 IEEE Conference on Local Computer Networks, pp. 102–108.
10. Zhu Y., Fu X., Graham B., Bettati R., and W. Zhao, “On flow correlation attacks and countermeasures in mix networks,” in PET04, LNCS 3424, 2004, pp. 207–225.

LIST OF PUBLICATIONS

1. S.Prince Sahaya Brighty, D.Sathya, “A Novel Approach for Detecting Wormhole Attack using Adhoc On-Demand Routing Protocol in MANET” International Conference on Innovations in Intelligent Instrumentation, Optimization and Signal Processing , Karunya University, Coimbatore, 1st and 2nd March 2013.
2. S.Prince Sahaya Brighty, D.Sathya, “Wormhole Attack Detection Using RC4 Algorithm in Mobile Adhoc Network” 5th National Conference on Computing,Communication and Information Systems, Sri Krishna College of Engineering and Technology, Coimbatore , 9th February 2013.