A Secure Routing Protocol for Mobile Adhoc Network

K.Rajesh kumr[1] and Dr.S.Radhakrishnan[2]
Department of Computer Science and Engineering, Kalasalingam University, Krishnankoil.
kumar85rajesh@gmail.com[1], srk@akce.ac.in[2]

Abstract:To secure a mobile ad hoc network (MANET) inadversarial environments, a particularly challenging problem is how to feasibly detect and defend possible attacks on routing protocols, particularly internal attacks, such as a Byzantine attack. In this paper, we propose a novel algorithm that detects internal attacks by using both message and route redundancy during route discovery. The route-discovery messages are protected by pair wise secret keys between a source and destination and some intermediate nodes along a route established by using public key cryptographic mechanisms. We also propose an optimal routing algorithm with routing metric combining both requirements on a node's trustworthiness and performance. A node builds up the trustworthiness on its neighboring nodes based on its observations on the behaviors of the neighbor nodes. Both of the proposed algorithms can be integrated into existing routing protocols for MANETs, such as ad hoc on demand distance vector routing (AODV) and dynamic source routing (DSR). As an example, we present such an integrated protocol called secure routing against collusion (SRAC), in which a node makes a routing decision based on its trust of its neighboring nodes and the performance provided by them. The simulation results have demonstrated the significant advantages of the proposed attack detection and routing algorithm over some known protocols**.**

1.INTRODUCTION

Mobile adhoc networks

As the popularity of mobile devices and wireless networks significantly increased over the past years, wireless ad hoc networks has now become one of the most vibrant and active fields of communication and networking research. Given many intriguing future applications, there are still some critical challenges and open problems to be solved.

QOS is a guarantee by the network to provide certain performance for a flow in terms of the quantities of bandwidth, delay, jitter, packet loss probability etc. Ad hoc networks make the appear an even more challenging problem than ever before, despite some of re-active routing protocols can be configured to return only paths that comply with certain desired parameters. Bandwidth is seriously limited. Our ultimate goal is to provide a model from the application layer to the MAC layer for supporting service differentiation. A transport layer protocol to support different data streams, queue management and a -supported MAC will be addressed in our future work.

The main challenges in assuring MANET networks are due to the fact that a mobile link is susceptible to attacks, and node mobility renders the networks to having a highly dynamic topology. The attacks against routing protocols can be categorized into external and internal attacks. An external attack originates from a router that does not participate in the routing process but masquerades as a trusted router. They can either advertise false routing information or generate floods of spurious service requests, such as a denial of service (DOS) attack. An internal attack originates from a compromised, misconfigured, faulty, or even malicious router inside a network domain. Among the internal attacks, *Byzantine attacks* can be defined as attacks against routing protocols, in which two or more routers collude to drop, fabricate, modify, or misroute packets in an attempt to disrupt the routing services.

II. RELATED WORKS

The current secure routing protocols for MANETs can roughly be divided into two categories, i.e., 1) those adding security mechanisms to the existing routing protocols and 2) those designed to detect and defend specific attacks. In the first category, the common practice is to secure the popular on-demand routing protocols, such as ad hoc on-demand distance vector routing (AODV), destination sequenced distance vector (DSDV), and dynamic source routing (DSR), by using a security association between the source and destination nodes such as pairwise secret keys and end-to-end authentication.

III. DYNAMIC KEY MANAGEMENT SCHEME AND ATTACK DETECTION ALGORITHM

We assume that a network is equipped with several security mechanisms in different layers in addition to the network layer. For example, the application layer can have some effective intrusion detection systems to monitor anomaly behaviors that can be used to detect and defend attacks such as DOS. In the network layer, the most possible attacks are data and routing information tampering. The majority of external attacks against routing protocols can be prevented by simple link layer encryption and authentication. We propose to have every node share a unique symmetric key with the source if it needs to transmit data. By applying this mechanism, the Sybil attack, the majority of selective forwarding and sinkhole attacks, and the HELLO flood attacks can be prevented. The major classes of attacks not countered are internal attacks and wormhole attacks. The defense mechanism for wormhole attacks can be found in. Therefore, we focus on internal attacks that are caused by authenticated routers, such as Byzantine attacks.

Dynamic Key Management Scheme

There are two basic key management approaches, i.e., public and secret key-based schemes. The public key-based scheme uses a pair of public/private keys and an asymmetric algorithm such as RSA to establish session keys and authenticate nodes. In the latter scheme, a secret key is a symmetric key shared by two nodes, which is used to verify the data integrity. Although a public key management system can be fully self-organized, the initial trust among the nodes in a network is still built by using external mechanisms. For example, Capkun *et al.* propose such a system by constructing a local certificate repository (CR) for each node . The initial construction starts by issuing public key certificates based on a users' own knowledge about other users' public keys. Initially, there is a PKI or CA to distribute the knowledge among users. Therefore, the work is a dynamic maintenance mechanism in building up the certificates.

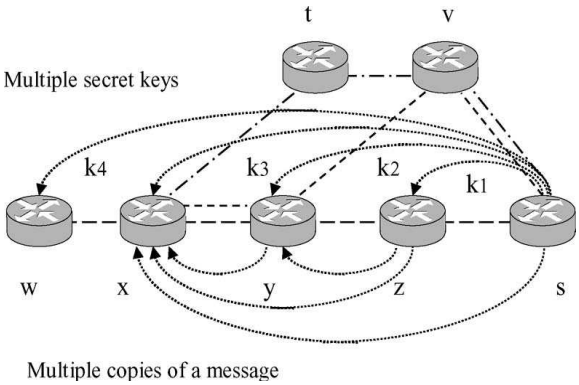

Fig.1 Demonstration of message and route redundancy.

Our framework for dynamic key management can be summarized as follows.

1) A secret key is established between the source and destination and some intermediate nodes along the route by using current public key information .
2) Each node along the route finds out which of its direct neighbors are faulty or compromised by using the established multiple keys between the source and intermediate nodes.
3) Each node updates its trustworthiness on each of its neighbors by using the observed node behavior and attack-detection results.
4) Each node constructs a local CR for the nodes it trusts. The certificates for those compromised nodes are immediately revoked. A node may

expand its CR by adding newly trusted nodes or exchanging repository information with its trusted neighbors.
5) By combining the current CR information and existing maintenance procedures for public key management, the nodes in the network can update public key information or build up a self-organized PKI

Route Discovery and Attack Detection

Based on the key management mechanism, the next task is to develop a framework for the secure discovery of the dynamic network topology. The attack detection scheme is incorporated into topology discovery procedures. Route discovery is straightforward for a node after it decrypts the received route discovery messages. To discover the routes in a dynamic environment, we need to use the inherent redundancies of the routes in ad hoc networks, called *route redundancy*, which means there are multiple, possibly disjoint, routes between nodes. As long as there are sufficiently many correct nodes, the routing protocols should be able to discover routes that go around some compromised nodes. Many ad hoc routing protocols such as AODV and DSR can discover multiple routes. Similar methods can be adopted into our scheme to discover multiple routes.

To detect internal attacks, including Byzantine attacks, we assume the following.
1) Each node has a pair of public/private keys and a unique ID. A compromised node participates in routing until detected.
2) The source and destination nodes are secured by external security agents. There is a shared key between the source and destination nodes.
3) Each of the intermediate nodes between the source and destination has established a shared key with the source node by using the key management scheme.
4) There are enough uncompromised nodes in the network so that a message can arrive at the destination via different routes.

Routing Algorithm

The heuristic algorithm can be summarized as follows.
1) During route discovery, a source node sends RREQ packets to its neighboring nodes. In these packets, along with the regular information, the node also sends its securityrelated information, such as key information.
2) Once an RREQ packet is received by an intermediate node, it calculates the TQI. The node places the link trustworthiness and QoS information in the RREQ packet and forwards it to its next hop. This process is repeated until it reaches the final destination.
3) At the destination, the node waits for a fixed number of RREQs before it makes a decision. Or else, a particular time can be set for which the destination or intermediate node needs to wait before making a routing decision. Once the various RREQs are received, the destination node compares the various TQI index values and selects the index with the least cost. It then unicasts the RREP back to the source node. When the source node receives the RREP, it starts data communication by using the route.
4) Once the route is established, the intermediate nodes monitor the link status of the next hops in the active routes. Those that donot meet the performanceandtrustworthiness requirements will be eliminated from the route.
5) When a link breakage in an active route is detected, a route error (RERR) packet is used to notify the other nodes that the loss of that link has occurred. Some maintenance procedures are needed as in AODV.

Simulation parameter:

Mac type	mac/802-11
Number of nodes	30
Number of packets	60
Packet size	1500 Mb
Bandwidth	11Mb
Slot time	50microsec
Packet interval	0.020 equal to send rate 8000 bytes

Protocol

In this section we present a routing protocol with Byzantine robustness and detection. Byzantine robustness means that the protocol routes packets from source to destination as long as a non-faulty path exists. Byzantine detection means that the protocol identifies faulty links. We first give a definition of what constitutes a faulty component and then justify this definition.

A faulty node is a node that:

- does not follow our protocol, or
- can be impersonated by another node.

The first part of the definition captures a node that is controlled by an adversary or executes buggy code. The second part of the definition is not obvious: we associate the notion of faulty with that of malicious or harmful but in this case,

the behavior of the faulty node does involve any malice. The faulty node can only be impersonated if, for example, its keys have been compromised. We cannot guarantee communication with a faulty node like this.

A faulty link is a link that:

- drops packets or
- is incident to a faulty node.

The first part of the definition is about links that have an impaired underlying communication system. Regarding the second part of the definition, we need to observe that a link that is incident to a faulty node can only route packets either from or to this node. If the faulty node has crashed, for example, then packets cannot be routed in either direction of the link. If the faulty node is a subverted one, then we would also like to avoid routing through this node, therefore identifying its incident links as faulty is equivalent from a routing robustness perspective to identifying this node as faulty. For performance reasons we would have liked to be able to identify faulty routers. However, we cannot tell with certainty whether a link or the downstream router is faulty, although we do not preclude certain cases where this can happen. Another reason is that a faulty router can invalidate its incident link without provision from the protocol. Therefore, if a link is detected to be faulty by our protocol, then one or more of the following statements are true:

- The upstream router is faulty.
- The underlying physical communication system is faulty.
- The downstream router is faulty.

The protocol can be seen as a combination of several components, each of which is important for the protocol's correctness.

These components are:

1. source routing,
2. destination acknowledgements,
3. timeouts,
4. fault announcements,
5. authentication,
6. reserved buffers,
7. sequence numbers, and
8. FIFO scheduling.

Attacks:

Attacks Using Modification

An attacker node may modify certain contents of the routing packet, thus propagating incorrect information in the network. Attacker node may change Sequence number or hop count in AODV.

Attacks Using Impersonation

A malicious node may try to impersonate a node and send data on its behalf. This attack is generally used in combination with modification attack. An attacker node may cause routing loops by sending fake RREP advertising higher sequence number, causing neighboring nodes to falsely update their routing tables.

Packet Dropping

Black Hole

An attacker may create a routing black hole, in which all packets are dropped. by sending forged routing packets, the attacker could route all packets for some destination to itself and then discard them.

Gray Hole

As a special case of a black hole, an attacker could create a gray hole, in which it selectively drops some packets but not others, for example, forwarding routing packets but not data packets.

Byzantine attacks

Byzantine attacks can be defined as attacks against routing protocols, in which two or more routers collude to drop, fabricate, modify, or misroute packets in an attempt to disrupt the routing services.

Network Simulator-2

NS-2 is a discrete event simulator targeted at networking research. Ns provides substantial support for simulation of TCP, routing, and multicast protocols over wired and wireless (local and satellite) networks. Ns is based in two languages, an object oriented simulator written in C++ and an OTcl interpreter, used to execute user command scripts. NS has a rich library of network and protocol objects. There are two class hierarchies the compiled C++ hierarchy and interpreted OTcl one, with one to one correspondence between them. The C++ compiled hierarchy allows us to achieve efficiency in the simulation and faster execution times. This is in particular useful for the detailed definition and operation of protocols. This allows one to reduce packet and even processing time. Then in the OTcl script provided by the user, we can define a particular network topology, the specific protocols and application that we wish to simulate and the form of the output that we wish to obtain from the simulator. The OTcl can make use of the objects complied in C++ though and OTcl linkage that create a matching OTcl object for each of the C++.

Simulation results:

Performance Evaluation:

The performance metrics are defined as follows:

1. Packet delivery ratio (PDR): The ratio of the total number of data packets successfully delivered to the destination to the total number of data packets sent out by a source node.

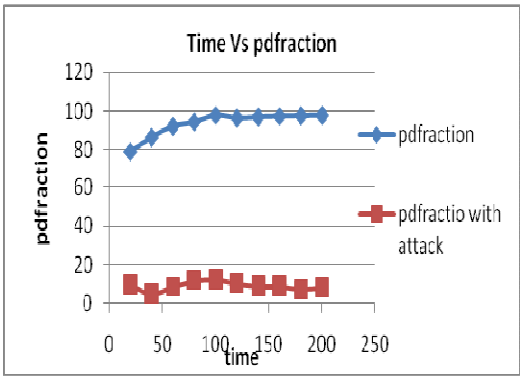

Fig.2 Pdfraction With and Without Attack

2. Average end to end delay:The average end-to-end delay of data packets is the interval between the data packet generation time and the time when the last bit arrives at the destination.

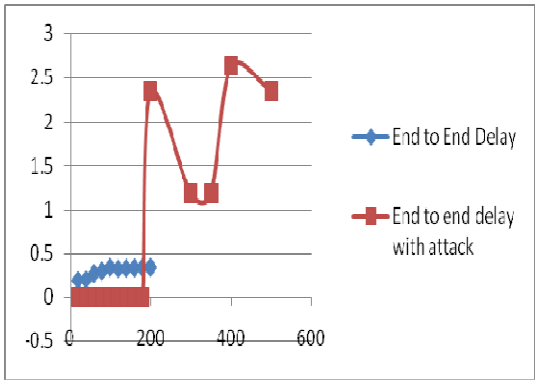

Fig.3 End to end delay With and Without Attack

3. Total throughput: The total number of data(application) packets that have been received at time *t* by a destination node.

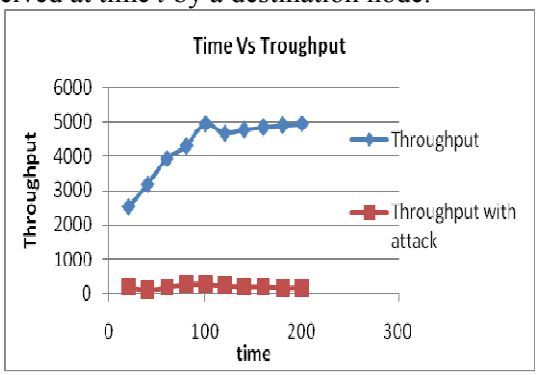

Fig.4 Troughput With and Without Attack

Conclusion

In this paper we implemented an AODV protocol that behaves as Black Hole in NS-2. We simulated five scenarios where each one has 20 nodes that use AODV protocol and also simulated the same scenarios after introducing one Black Hole Node into the network. Moreover, we also implemented a solution that attempted to reduce the Attack effects in NS-2. AODV network has normally 3.21 % data loss and if a Attack Node is introducing in this network data loss is increased to 92.59 %. As 3.21 % data loss already exists in this data traffic, Attack Node increases this data loss by 89.38 %.

Future work

We simulated the Black Hole Attack in the Ad-hoc Networks and investigated its affects. In our study, we used the AODV routing protocol. But the other routing protocols could be simulated as well. All routing protocols are expected to present different results. Therefore, the best routing protocol for minimizing the Black Hole Attack may be determined. There are many Intrusion Detection Systems (IDS) for ad-hoc networks. These IDSs could be tested to determine which one is the best to detect the Black Hole.

ACKNOWLEDGEMENT

First of all we thank the almighty for giving us the knowledge and courage to complete the research work successfully. We express our gratitude to our respected Vice Chancellor **Dr. Chelliah Thangaraj** M. Tech., Ph.D., for allowing us to do the research work internally. Also we acknowledge the support provided by TIFAC-CORE Network Engineering, Kalasalingam University (Supported by Department of Science and Technology, Government of India).

REFERENCES

[1] A Secure Routing Protocol Against ByzantineAttacks for MANETs in Adversarial Environments, Ming Yu, Senior Member, IEEE, Mengchu Zhou, Fellow, IEEE, and Wei Su, Senior Member, IEEE., IEEE Transactions on Vehicular Technology, Vol. 58, no. 1, january 2009

[2] P. Papadimitratos and Z. Haas, "Securing the Internet routing infrastructure," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 60–68, Oct. 2002.

[3] C. Karlof and D. Wagner, "Secure routing in wireless sensor networks: Attacks and counter - -measures," *Ad Hoc Netw.*, vol. 1, no. 2/3, pp. 293–315, Sep. 2003.

[4] C. Zhang, M. C. Zhou, and M. Yu, "Ad hoc network routing and security: A review," *Int. J. Commun. Syst.*, vol. 20, no. 8, pp.909– 25, Aug. 2007.

[5] J.-S. Hwu, R.-J. Chen, and Y.-B. Lin, "An efficient identity-based cryptosystem for end-to-end mobile security," *IEEE Trans.Wireless Commun.*,vol. 5, no. 9, pp. 2586–2593, Sep. 2006.

[6] C. E. Perkins and E. M. Royer, "The ad hoc on-demand distance-vector protocol," in *Ad Hoc Networking*, C. E. Perkins, Ed. Reading, MA: Addison-Wesley, 2001, ch. 6, pp. 173–220.

[7] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proc. ACM SIGCOMM Conf. Comms. Architectures, Protocols Appl.*, 1994, pp. 234–244.