A Secure Routing Protocol for Wireless Ad Hoc Networks

Huaizhi Li
Department of Computer Science
University of Kentucky
Lexington, KY 40506
Email: hli3@cs.uky.edu

Mukesh Singhal
Department of Computer Science
University of Kentucky
Lexington, KY 40506
Email: singhal@cs.uky.edu

***Abstract*— Ad hoc networks, which do not rely on infrastructure such as access points or base stations, can be deployed rapidly and inexpensively even in situations with geographical or time constraints. Ad hoc networks have attractive applications in both military and disaster situations and also in commercial uses like sensor networks or conferencing. However, the nature of ad hoc networks makes them vulnerable to attacks, especially in the routing protocol. How to protect an ad hoc routing protocol is an important research topic. In this paper, we present an on-demand secure routing protocol for ad hoc networks based on a distributed authentication mechanism. The protocol makes use of recommendation and trust evaluation to establish a trust relationship between network entities and uses feedback to adjust it. The protocol does not need the support of a trusted third party and can discover multiple routes between two nodes.**

I. Introduction

An ad hoc network is a set of wireless mobile nodes that form a dynamic autonomous network without the intervention of centralized access points or base stations. Unlike traditional wireless networks, ad hoc networks require no fixed network infrastructure and can be deployed as multihop packet networks rapidly and with relatively low expense. Such networks can be very useful in scenarios where natural conditions or time constraints make it impossible to pre-deploy infrastructure. Examples of applications include battlefields, emergency services, conference rooms, and home and office.

Mobile nodes in an ad hoc network have limited radio transmission range. Nodes that are unable to communicate directly with each other require that intermediate nodes forward packets for them. Each node acts both as a router and as a host. The function of a routing protocol in ad hoc network is to establish routes between different nodes. Several ad hoc routing protocols have been proposed, which include AODV [1], [2], DSR [3], ZRP [4], TORA [5], DSDV [6], TBRPF [7], and others. Although they represent important steps in ad hoc routing research area, they still have security vulnerabilities, and can be attacked [8]. The special characteristics of ad hoc networks put forward challenges not present in traditional wired networks. In the traditional Internet, routers within the central parts of the network are owned by a few well known operators and are therefore assumed to be more trustworthy [9]. This assumption no longer holds in an ad hoc network since neither centrally administrated secure routers nor a strict security policy exists in an ad hoc network, and all nodes entering the network are expected to take part in routing. Also, because the links are usually wireless, any security that was gained because of the difficulty of tapping into a network is lost. Furthermore, because the topology in such a network can be highly dynamic, traditional routing protocols can no longer be used. Thus an ad hoc network has much higher security requirements than the traditional networks and the routing in ad hoc networks is hard to accomplish securely, robustly and efficiently.

The general purpose of securing ad hoc routing protocols is to protect the routing messages, to prevent attackers from modifying these messages or even injecting harmful routing messages into the network. So integrity and authenticity of routing messages should be guaranteed. Confidentiality can be ensured easily, e.g., by encryption, but it will increase overhead. Route establishment should be a fast process. If too much security mechanisms are built in, the efficiency of routing protocol may be sacrificed. So there is a tradeoff between security and efficiency. In ad hoc network, the network topology is dynamic. Different packets exchanged between the same two nodes may go through different routes, among which there may be attackers lurking. Nevertheless, without online trusted servers as in wired networks, it is difficult to be acquainted with the trustworthiness of each node, thus keeping away malicious nodes from the routes.

The common approaches to fight against attacks are cryptographic algorithms: encryption/decryption algorithms [10] ensure data secrecy and prevent eavesdropping attack; digital signature schemes [10] provide authenticity and integrity and prevent modification and impersonation attacks. These approaches are also called "hard security" mechanisms [11] or prevention strategy [12]. Cryptographic schemes are effective to fight against attacks, but are not able to prevent selfishness like misbehaviors. For example, a node may refuse to forward data packets for other nodes to save its battery. So a comprehensive approach is necessary for ad hoc networks to prevent both attacks and misbehaviors. In this paper, we combine the "hard security" mechanisms with dynamic trust management and propose a new secure routing protocol for ad hoc networks. The secure routing protocol is based on a "Distributed Authentication Model" whose mechanism is

0-7695-2507-5/06/$20.00 (C) 2006 IEEE

trustworthiness acquiring and adjusting of network nodes with no online trusted servers. With this Distributed Authentication Model, our protocol can exclude the attackers and selfish nodes timely and proactively. Message integrity is ensured by Message Authentication Code [13]. Moreover, it can set up multiple paths between two nodes and copes with dynamically changing topology.

The paper is organized as follows. Section II briefly discusses possible attacks to ad hoc network routing protocols. Section III describes our Distributed Authentication Model. Section IV presents our secure ad hoc routing protocol. Section V gives an analysis of our protocol. In Section VI, we present existing work related to our protocol. The final section concludes the paper and points out future research directions.

II. Attacks to Ad Hoc Network Routing Protocols

Our main concern is those attacks, which are trying to improperly modify data, gain authentication, or gain authorization by inserting false packets or by modifying packets. The attacks analyzed here are general attacks to ad hoc network routing protocols, not to a specific routing protocol. Broadly speaking, attacks to routing protocols come mainly from two sources, external and internal [8]. External attacks come from outside intruders who do not belong to the network. Internal attacks come from compromised nodes in the network.

A. External Attacks

An outside intruder could attack a routing protocol in various ways. Specific threats include the following:

- Replay attack: An intruder could passively collect routing information, for example route request/reply. Later, the intruder could retransmit "obsolete" routing information. If obsolete information is accepted and disseminated, a node could make incorrect routing decision.
- Denial of Service (DoS): A malicious node could generate false routing messages and flood them into the network so that a portion of network resource is wasted by these junk messages and some CPU cycles and memory of nodes are taken up by the processing of them. Sometimes, DoS attacks are quite harmful. For example, an attacker can broadcast bogus route error messages about some links so that these links are thought to be down.
- Modification: Malicious nodes can modify fields of a routing message, like sequence number and hop counts of AODV, to cause redirection of network traffic. If integrity measures, e.g., Message Authentication Code, are used to protect the routing message, this attack will not succeed, but it is still a DoS attack.
- Masquerading: A malicious node can launch IP spoofing attack, impersonate other nodes, and disseminate false routing message so that the routing tables are not consistent.

TABLE I
Alice's Trust Table

Entities	Trust Values	Trustworthy or Not
Bob	2.2	yes
Cathy	3.1	yes
David	1.5	no

B. Internal Attacks

Internal attackers may control everything of the compromised nodes, even the private keys or shared secrets with other nodes. An internal attacker can launch all the attacks of an external attacker and it is more harmful. Even with security measures, the internal attackers can still pass authentication and generate correct Message Authentication Code for modified or fabricated routing updates. So internal attacks are difficult to detect and handle.

When an internal attacker inhabits in a network, it has two choices during the route discovery process: launching attacks to destroy routing infrastructure, and behaving as a benign node during route establishment and launching attacks during data packet delivery [14]. So in either case the attacker does not want to be excluded from the routing protocol, otherwise it gains no benefit.

III. A Distributed Authentication Model

In our protocol, authentication among different nodes is done by a "Distributed Authentication Model", which is described in this section.

In an ad hoc network, each node maintains a repository (Trust Table) of known entities. Each entity on the table is assigned a trust value depending on its reliability. The trust value metric is an important factor on the accuracy of the trust management system. At present there are no standards [11]. We utilize a concrete trust value metric. The trust value of a node can be: -1(distrust), 0(ignorance), 1(minimal), 2(average), 3(good), and 4(complete), where the number is the trust value and the word in "()" gives the meaning of the value. In our protocol, as long as an entity's trust value is ≥ 2, it is assigned a "yes", meaning "trustworthy", otherwise, it is assigned a "no", meaning "untrustworthy". As an example, Alice's Trust Table is shown in table I.

In the following sections, we present the components of our Distributed Authentication Model.

A. Trust Value Query

When a node (A) authenticates another node (B), node A first checks its own Trust Table. If node B is in its table and the value is "yes", then B can be trusted; if the value is "no", B can not be trusted. If B is not in A's Trust Table, A sends a trust-value-request for B's trust value to all the trustworthy nodes in its Trust Table. If any of these trustworthy nodes does not know B, this node passes A's request to its trustworthy nodes in its Trust Table. So this is a recursive process (cycle in recommendation should be prevented) and eventually this trust-value-request may reach a node, e.g., node

C, which knows B's trust value. The node that sends the trust-value-request to node C, e.g., node E, has node C in its Trust Table. Node C sends a trust-value-reply to A along the reverse path that the trust-value-request travels. The trust-value-reply carries node B's trust value. When the trust-value-reply reaches node E, node E puts node C's trust value into the trust-value-reply and forwards the reply to node A. Node A may receive several replies from other nodes. Next, node A calls a "Trust Evaluation" process to evaluate node B, which is described in the next section.

B. Trust Evaluation

The evaluation process allows A to decide if B can be trusted or not. It processes the received data and outputs a node's trust value. The following function is used to calculate B's trust value:

$$trust\_value(T) = \frac{\sum_{i=1}^{n}(trust\_value(i) \cdot trust\_value\_rp(i))}{\sum_{i=1}^{n} trust\_value(i)}$$

where:
n : the number of replies received.
$trust\_value(T)$: the trust value of a target node, in this example, node B.
$trust\_value\_rp(i)$: the ith trust value returned.
$trust\_value(i)$: the trust value of the node that returns $trust\_value\_rp(i)$.

In this example, assume that node C whose trust value is 3 is the first node to send back node B's trust value which is 2.5 to A, then $trust\_value\_rp(1)$=2.5 and $trust\_value(1)$=3.

Basically, this is a weighted average with the trust value of the node that replies as weight. With the computed trust value of B, A can decide whether B is trustworthy or not and assign it a "yes" or "no" and adds B to its Trust Table.

C. Feedback

If A receives a "yes" about B (trust value is ≥ 2) from node C, but later A finds itself cheated by B, A can inform C about its wrong information and decrease the trust value of C in its Trust Table. A might also put B and C on a list of suspicious nodes. C can take some actions to find the problem. If C finds B misbehaving itself, C can send a special message to A with negative information about B. A may adjust B's trust value depending on how reliable C is. This message can also be broadcast.

D. Trust Monitor

The behavior of a node changes overtime. A trustworthy node may become malicious later. It is important to monitor the performance of network entities and adjust their trust value timely. We utilize a mechanism similiar to watchdog [15]: each node monitors its neighbors. If a neighbor is found misbehaving, e.g., dropping data packets, its trust value is decreased. When a node's trust value is below a threshold, like 2, it is considered as a malicious node and put on a black list. A WARNING message is also generated and disseminated in the network to notify other nodes the detected malicious node. When a node receives the WARNING message, it checks if the originator of the WARNING message is trustworthy or not. If it is trustworthy, the node adds the malicious node to its black list. Otherwise, the node discards the WARNING message.

E. Digital Signature

If public key cryptography is implemented in the network and each node's public key is distributed by using approaches like [8], [16], [17], a node may require the nodes that reply its trust-value-request to sign their trust-value-replies so that non-repudiation is ensured. The WARNING messages in Section III-D can also be protected by digital signature. Because public key cryptography is very expensive, we limit its usage and we do not use public key cryptography to protect routing messages, like route request and route reply.

IV. The Secure Routing Protocol

In this section we present our secure routing protocol for ad hoc networks. It is an on-demand routing protocol and satisfies the special features common to ad hoc networks: dynamically changing topology and low-power devices. We do not assume the existence of trusted servers, which may be infeasible for ad hoc networks. The protocol can discover multiple paths between two nodes. This is essential for an ad hoc network to be able to tolerate attacks induced path failures and provide robust packet delivery [8], [14]. Our protocol is designed for the following situations: (1) military applications, e.g., battle field. (2) emergency situations. The sole assumption of the protocol is that at the beginning, all the nodes share a group key K and can be trusted. This is a reasonable assumption since all the members belong to same troop or team. Certainly it is possible that some nodes may be compromised later and become untrustworthy. Yet as analyzed in Section V, these attacks can be prevented or detected. Moreover, our protocol can keep out the attackers timely and proactively. We do not consider physical layer and link layer attacks in this paper, like jamming attacks. We also presume that there are no collaborating attacks.

A. Setup

Every node installs the Distributed Authentication Model. Each node creates a Trust Table. At the begining, all the nodes can be trusted and are assigned a trust value. An optimistic node might assign a larger trust value to other nodes, e.g., 4, while a conservative node may assign a smaller value, e.g., 3. This is the starting point of trust.

B. Neighbor Discovery and Key Establishment

Every node periodically broadcasts a HELLO message. With this mechanism, a node can detect its new neighbors. When a new neighbor is found, a node invokes the Distributed Authentication Model to authenticate the neighbor and puts the neighbor in its Trust Table.

Every node sets up a secret key with each trustworthy neighbor by using a two-party key establishment protocol [18].

C. Route Discovery

Like other on-demand routing protocols, route discovery consists of route request process and route reply process, which are described next.

1) Route Request: The source node generates a route request (RREQ) and attaches to it a Message Authentication Code (MAC). The MAC covers the whole route request message and is generated by using a keyed hash algorithm [13] and the shared group key K (to prevent modification from external attackers). A RREQ contains the following fields: source IP address, destination IP address, a sequence number, and a MAC:

$$RREQ : \{IP_d, IP_s, Seq\_num\}||K(MAC) \quad (1)$$

Where, IP_d and IP_s are IP addresses of destination and source nodes. "||" means concatenation and K(MAC) denotes a MAC generated by using key K. The sequence number is maintained by the source node for each destination node. The sequence number increases monotonically for each route request. The message RREQ is broadcast.

In our secure routing protocol, each intermediate node maintains a route request table (RREQ-Table), which stores the first route request with a specific Seq_num originated from a particular source node and received from a trustworthy node. All later received same RREQ are dropped. The destination node processes the same RREQ received from different nodes. When a node receives a RREQ and it is not the destination node, it checks if the route request is a new route request from that source node. If not, the node drops the RREQ. Otherwise, the node verifies the MAC of the RREQ using the shared group key K. If the MAC is correct, the node authenticates the sender of the RREQ using the Distributed Authentication Model. Using the ad hoc network in Figure 1, we explain how this is done.

In Figure 1, there exist multiple routes between the source node S and the destination node D. Let's focus on the path with dotted line first to see how this route is established. After B receives an RREQ from S and verifies the MAC, B tries to authenticate S. B first checks its Trust Table. If S can be trusted, B accepts the RREQ, stores the RREQ in its RREQ-Table, and rebroadcasts the RREQ. If S can not be trusted, B drops the RREQ. A RREQ generated by a malicious node is ignored. If S is not in B's Trust Table[1], B uses the Distributed Authentication Model to check S's trustworthiness and sets a timeout value. If the result is "no" or timeout occurs and B has not received any replies, B drops the RREQ. If the result is "yes", B stores the RREQ in its RREQ-Table and rebroadcasts the RREQ. Then B establishes a secret key with S using a two-party key establishment protocol [18]. When node E receives the RREQ, it verifies the freshness of the RREQ and the MAC, and authenticates B. If the verification is passed and B is trustworthy, E stores B as the next hop to the source node S in its routing table, stores the RREQ in its RREQ-Table, and rebroadcasts the RREQ. If B is not trustworthy, the RREQ is dropped. Node G follows the same procedure as E. Finally, D receives the RREQ from G. If the MAC is correct and G can be trusted, D stores G as the next hop to the source. So a path is detected from S to D.

[1]In general this rarely happens. The neighbor discovery procedure (Section IV-B) detects and authenticates new neighbors timely.

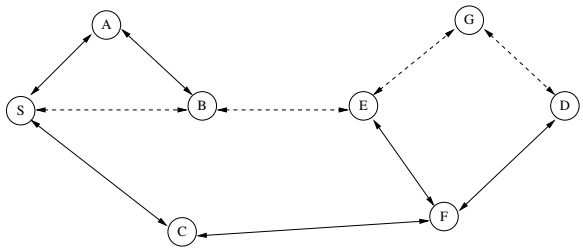

Fig. 1. An example ad hoc network. S-source node; D-destination node; A, B, C, E, F and G are intermediate nodes.

Node B may also receive the same RREQ from node A. Since B already received the RREQ from S, B drops the RREQ received from A.

There are two other possible paths from S to D, S-B-E-F-D and S-C-F-D, which depend on node F. If node F receives the RREQ from node C earlier than the RREQ from node E and C is trustworthy, F stores C as the next hop to the source node S and stores the RREQ in its RREQ-Table. The RREQ from E is dropped. When the destination node D receives the RREQ from F, it accepts the RREQ and stores F as the next hop to the source. So the path S-C-F-D is detected. On the contrary, if node F receives the RREQ from node E first, F stores E as the next hop to the source node S and the RREQ from node C is dropped. The path S-B-E-F-D is detected.

Suppose node E is trustworthy and node C is not trustworthy. If F receives the RREQ from C earlier than from E, F drops the RREQ received from C and accepts the one from E.

After the above procedure, all the nodes involved are authenticated and a secret key is set up between each pair of neighbor nodes. The untrustworthy or suspicious nodes are excluded.

2) Route Reply: The destination node D stores two nodes (G and F) as the next hop to the source node S. Node D generates two route replies (RREP) and attaches to them a MAC calculated by using the secret key that D shares with G and the key that D shares with F, respectively. The RREPs are:

$$RREP : \{IP_s, IP_d, Seq\_num\}||K_{DG}(MAC), \text{for node G} \quad (2)$$

$$RREP : \{IP_s, IP_d, Seq\_num\}||K_{DF}(MAC), \text{for node F} \quad (3)$$

where, K_{DG} is the key shared between D and G and K_{DF} is key shared between D and F.

D unicasts the RREPs to G and F separately.

When G receives D's RREP, G checks if it stores the corresponding RREQ. If not, G discards the RREP. Otherwise, G verifies the MAC. If it is correct, G updates the entry for D in its routing table. G replaces the MAC in the RREP by a

MAC calculated using a secret key that it shares with E. The updated RREP is:

$$RREP : \{IP_s, IP_d, Seq\_num\}||K_{GE}(MAC) \quad (4)$$

Then G unicasts the RREP to E. When E receives the RREP, it follows the same process as node G: verify the MAC, store G as the next hop to the destination node in its routing table, and unicast the RREP to node B. The RREP is protected by a MAC calculated by using the key that E shares with B. Finally, the source node S receives the RREP. S verifies the MAC and stores the node B as the next hop to destination D in its routing table.

When node F receives D's RREP, it follows the same process as node G. There are two cases: if the detected path is S-C-F-D, F unicasts the RREP to C. The RREP is protected by a MAC calculated by using the key shared by F and C. The source node receives the RREP from node C and stores C as the next hop to the destination node D. If the detected path is S-B-E-F-D, F unicasts the RREP to E. The RREP is protected by a MAC calculated by using the key that F shares with E. When node E receives the RREP, it verifies the MAC and stores F as the next hop to the destination node in its routing table. Now E has two paths to the destination, one through G and one through F. Node E does not forward the second RREP to node B. In this case, node B and S only receive one RREP and only store one path in their routing tables.

Eventually, two paths are established between S and D:

- case 1: S-B-E-G-D and S-C-F-D
- case 2: S-B-E-G-D and S-B-E-F-D

When node S sends data packets to node D, at each hop, a node randomly picks a path if multiple paths exist, which can balance the load. If the shortest path is always used, it may cause congestion. For case 1, S selects either C or B as the next hop. For case 2, S sends the data packet to B, since S only knows one path to D. When B receives the data packet, it forwards the packet to E, since B also knows one path to D. When E receives the data packet, E picks either G or F as next hop to forward the packet.

3) Route Maintenance: When a link is broken, the node upstream the link checks the broken routes caused by the broken link. If there are some destination nodes, to whom all the routes are broken, the node generates and broadcasts a route error message which contains those broken routes. If to some destination nodes, there are still other routes available, the broken routes to these destinations are not put in to the route error message. For example, in Figure 1, for case 2 in Section IV-C.2, if the link B-E is broken, node B generates a route error message; if link E-F is broken, node E does not generate a route error message, since the path E-G-D is up. When both the links E-G and E-F are broken, E generates a route error message, which contains the broken routes. To protect the route error message, the originator of the error message attaches a MAC to the error message, which is calculated by using the group key K. When a node receives the route error, it authenticates the originator of the error message, if the originator is not trustworthy, it drops the error message. Otherwise it verifies the MAC and invalidates the routes contained in the error message, if it has those routes in its routing table.

4) Attacker Isolation: As discussed in Section III-D, whenever a node finds that the trust value of one node (e.g., node T) in its Trust Table is decreased to less than 2 (means untrustworthy), it removes all the routes containing node T (T is the next hop or the destination) from its routing table and broadcasts a WARNING message to inform all other nodes that T is suspicious. When a node receives a WARNING message, it first authenticates the trustworthiness of the originator of the WARNING message. If the originator is trustworthy, it updates T's trust value and removes all the routes containing T (T is the next hop or the destination) from its routing table. Otherwise, the node just ignores the WARNING message. With this mechanism, the protocol can exclude attackers timely and proactively.

D. Protection of Data Packet

As analyzed in Section II, an internal attacker may behave as a good node and participate in route establishment during route discovery process, and attack data packet during real data communication process. So it is necessary to provide comprehensive protection for routing protocol and data traffic, especially for ad hoc networks. The Distributed Authentication Model provides protection to data packet communication. For example, when a node finds another node drops packets, it may consider that node suspicious and decrease its trust value so that during later route discovery that node might be excluded. Integrity of data packets can be protected. When a data packet travels from a source node to a destination node, each hop calculates a Message Authentication Code by using the key that it shares with the next hop and attaches the Message Authentication Code to the data packet. When the next hop receives the data packet, it can verify the Message Authentication Code to check the integrity of the packet.

V. Analysis of the Protocol

In this section, we appraise the overhead and security of our protocol.

A. Performance and Overhead Analysis

We evaluated the performance of our secure routing protocol using ns2 simulator [19] and the wireless extensions developed by CMU. In the experiments, the MAC layer is the IEEE 802.11 MAC protocol with Distributed Coordination Function (DCF) [20], which uses Request-to-send (RTS) and Clear-to-send (CTS) control frames for unicast packet. The characteristics of radio model is similar to Lucent's WaveLAN [21], which is a shared-media radio with 2Mb/sec transmission rate. We simulated 100 nodes in a 1500m $\times$ 1500m area. We used 20 source-destination pairs. The source and destination nodes are randomly selected. Traffic sources are CBR (constant bit-rate). Each source sends data packets of 512 bytes at the rate of four packets per second. The simulation time was 900

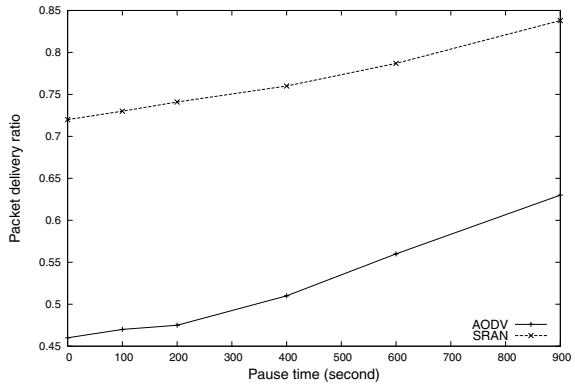

Fig. 2. Packet delivery ratio.

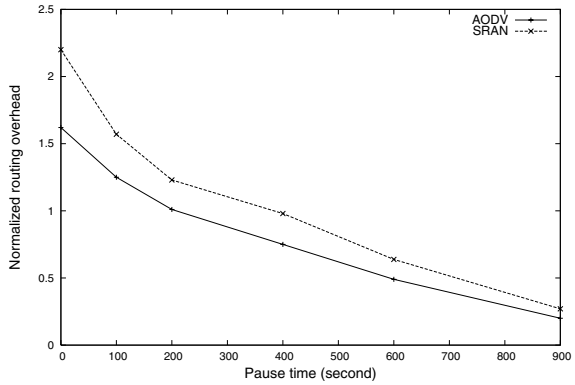

Fig. 3. Normalized routing overhead.

seconds. The random way point model [22] was used as the mobility model. The maximum node speed was 20 m/s and the minimum speed was 1 m/s. We varied pause time to change mobility rate. The pause time was 0, 100, 200, 400, 600 and 900 simulated seconds.

Similar to SEAD [23] and Ariadne [24], we evaluted the performance of our secure routing protocol without attackers. We assume that 20 of 100 nodes are misbehaving nodes, which participate in route discovery, but drop all the data packets that they should forward. The 20 misbehaving nodes are randomly selected, but are neither source node nor destination node. We compare the performance of our secure routing protocol with AODV [1]. Identical traffic and mobility scenarios are used for both the routing protocols.

We use the following metrics to measure the performance:

- *Packet delivery ratio:* the number of data packets received at the destinations to the number of data packets generated by the CBR sources.
- *Normalized routing overhead:* the number of routing messages transmitted in the network for each data packet received at the destination.

Figure 2 and 3 show the simulation results, in which SRAN represents our secure routing protocol. From Figure 2, it can be seen that SRAN achieves about 50 percent higher packet delivery ratio than AODV. This is explained as follows. Since the misbehaving nodes participate in the route discovery process, they forward all the routing messages, like route request and route reply. So in AODV, a path between a source and a destination node may contain misbehaving nodes, which drop all the data packets that they should forward, thus degrade the packet delivery ratio. SRAN can detect those misbehaving nodes and avoid them during route discovery. Furthermore, SRAN is a multi-path routing protocol. If one route is broken, a backup route may exist to continue forwarding the data packets, thus avoid packet dropping.

Figure 3 shows the comparison of routing overhead. The normalized routing overhead metric includes all the control messages to establish route, such as route requests and route replies. For SRAN, it also includes the control messages for the Distributed Authentication Model and key establishment, which cause SRAN to have higher overhead than AODV. For secure routing protocols, the overhead caused by security mechanisms is unavoidable. For SRAN, the overhead of the Distributed Authentication Model mainly concentrates in the setup stage. During this stage, each node discovers its neighbors, establishes secret keys with them, and constructs a Trust Table. If the network topology is relatively stable, the computational work that an intermediate node needs to do is just Trust Table look up, which is very fast. If the network topology changes, a node may have new neighbors. The node needs to authenticate the new neighbors by using the Distributed Authentication Model, and set up secret keys with them (Section IV-B). However, this small expenditure will be amortized: route discovery in later rounds will be faster since more nodes have established trust relationship between each other, so the authentication is faster. Moreover, SRAN is a multi-path routing protocol. One round of route discovery can establish multiple routes from a source to a destination. If one route is broken, a backup route can be used. For AODV, One round of route discovery can only discover one route from a source to a destination. If the route is broken, a route rediscovery may be needed, which incurs overhead.

Each node maintains a Trust Table and stores the secret keys shared with its trustworthy nodes. This takes up some memory and it is proportional to the number of nodes in the network. As an example, if there are one thousand nodes in the network and each secret key is 128 bits long, a node needs less than 16 KB memory to store the keys. If each entry in the Trust Table is 32 bytes, the Trust Table requires 32 KB memory. So the total memory consumption of each node is no more than 48 KB (it should be much less than 48 KB, since in general a node does not need to establish a key with all the nodes). Certainly, each node also maintains a routing table, which is a necessary component of all the routing protocols. Therefore the memory overhead is low.

B. Security Analysis

We now give an analysis of the security properties of our protocol.

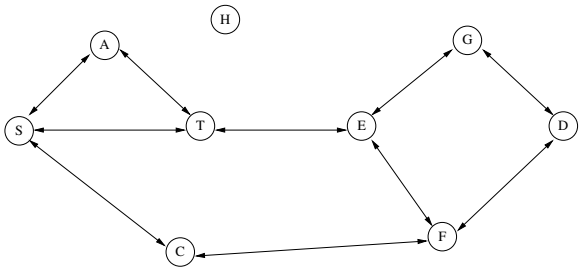

Fig. 4. An example ad hoc network. S-source node; D-destination node; T-internal attacker; A, C, E, F, G and H are intermediate nodes.

1) External Attacks: Fabrication from external attackers is prevented, because the external attackers are excluded by the Distributed Authentication Model. An external attacker may try to modify a routing message. Since external attackers do not possess the group key K or the secrets shared between each pair of nodes, they can not generate correct MAC corresponding to the message. So modifications can be prevented. IP spoofing can be prevented too. There are two possibilities. First, an attacker may impersonate a nonexistent node, but the Distributed Authentication Model will keep it out. Second, the attacker may impersonate an existent node, e.g., node A in Figure 1, and replay an eavesdropped route request. When node B receives this route request, it will accept the message if it has never received it. But when B receives the corresponding route reply later, B will forward the route reply to A rather than the attacker. Thus node A will detect the problem if node A is in the transmission range of node B. If A is out of B's transmission range, the route reply is lost. Simple replay attacks can be prevented by sequence number.

An external attacker may drop the received RREQs and RREPs and refuse to forward them. It can not be prevented. But this excludes the attacker from the route. This kind of denial-of-service attack exists for all the security protocols.

2) Internal Attacks: If a node is compromised, then the attackers may control everything at the node: the shared group key K, the shared secrets with other nodes, and the Trust Table. The attacker can pass authentication and generate correct MAC. For example, suppose node B in Figure 1 is compromised and becomes node T as shown in Figure 4. Other nodes in Figure 4 are good nodes. Possible attacks are analyzed below.

a) Attacks to Route Request (RREQ): When attacker T receives an RREQ from A, the following attacks are possible:

- IP spoofing attack
  When forwarding RREQ to E, T may put an incorrect IP address in the message. T has two choices: (1) using nonexistent IP address, which will be detected by E, since it can not be authenticated; (2) using existing IP address, e.g. IP address of node H. Node E will accept it, if H is in E's Trust Table and can be trusted. Otherwise E will try to authenticate H and set up a key with H. This requires H to be in E's transmission range. If not, E will drop the RREQ and the attack fails. But even if E accepts the RREQ, when E receives the corresponding RREP from G or F, E will forward it to H rather than T. There are three possibilities: (1) H is not participating in the current route discovery. H will find the problem, since it never forwards the RREQ. (2) H is a participator and within the transmission range of E. The attack helps route discovery other than does something harmful, since H should receive a RREP from E. (3) H is a participator and not in the transmission range of E, the RREP will be lost, as H can not receive it. In all, the attack fails.
- Modification
  Attacker T can modify the source address (IP_s) of RREQ and generate a correct MAC. When E receives the modified RREQ, E will not detect it. The destination node D will not detect it either. When T receives the RREP from E, T modifies it back to the correct one and then forwards it to A. The final result is still a correct path from S to D, although is modified during the process of route discovery. So the attacker is not able to disrupt the route discovery. An attacker can modify the destination address (IP_d) of RREQ. If IP_d is modified to a nonexistent IP address, eventually the RREQ will disappear. If T modifies IP_d from the IP address of D to IP addresses of some other node, e.g., IP address of E. Node E will not find the problem. When T receives the corresponding RREP from E, T modifies IP_d back to the IP address of node D, and forwards to A, A will accept it. So until now the attack seems successful. But after T modifies the RREQ, it needs to broadcast the modified RREQ. When other nodes receive it, they will rebroadcast it. Due to the flooding of RREQ, the original source of RREQ, i.e., node S, will receive the modified RREQ, and find the problem.

b) Attacks to Route Reply (RREP): When attacker T receives RREP from E, the following attacks are possible:

- IP spoofing
  When T forward the RREP to A, T uses some other nodes' IP addresses instead of using its IP address to poison A's routing table. If T uses a nonexistent IP address, A will detect it because there is no key shared between A and the nonexistent IP address. If T wants to use the IP address of an existing node, e.g., IP address of node H, it requires that H is in A's Trust Table, and T needs to break the secret key shared between A and H. This makes the attack difficult to succeed.
- Modification
  Attacker T can modify the source and destination addresses (IP_s and IP_d) of RREP. But node A and S will find the problem and drop the modified RREP, because there is no corresponding RREQ.

c) Modification of the Sequence Number of RREQ: Attacker T can modify the Seq_num of a received RREQ to a larger value. Other nodes will accept it. But this will cause the future real RREQ to be dropped, because its Seq_num is less than the stored one. However, this will also exclude the

attacker since no route will be established. Furthermore, the original source node will also receive the flooded modified RREQ and detect the problem.

An internal attacker may spoof victim V and send a RREQ with the maximum sequence number to cause V's subsequent route requests to be dropped. But due to the flooding nature of RREQ, node V will receive that RREQ and detect the problem.

d) Fabrication of Route Reply: When attacker T receives a route request, T can forge a route reply instead of forwarding the route request to other nodes. T can copy the source and destination addresses and the sequence number from the route request. Since T knows the secret keys that it shares with A and S, T can generate the correct MAC for the route reply. So T is able to fabricate a correct route reply and send it to A and S. Nodes A and S will not detect the problem. They will think that the route replies are forwarded by T and accept it. So an invalid path is established. But our protocol discovers multiple paths, so there are still other paths to be used. If there is a secret key shared between the source and destination nodes, this problem can be prevented by utilizing a MAC calculated using the secret key. Stajano's "resurrecting duckling model" [25]–[27] can also be used.

3) The Shared Group Key: To further improve the security of our protocol, a group key establishment protocol, like [28], may be utilized to update the shared group key K. When a malicious node or an attacker is detected, the group key establishment protocol is executed to update K, which excludes the malicious node and ensures perfect forward secrecy. Certainly, the updating process will cause some overhead.

4) Message Authentication: Our protocol makes use of pair-wise authentication. The routing messages are authenticated by each pair of neighboring intermediate nodes. For RREQ, two neighboring nodes authenticate each other through the Distributed Authentication Model. For RREP, a MAC generated by using the secret key shared between two neighboring nodes provides message authenticity.

If the destination node is in the Trust Table of the source node or vice versa, the two ends already establish a shared secret key. End-to-end authentication can also be utilized besides pair-wise authentication. Each routing message carries two MACs: one for pair-wise authentication; one for end-to-end authentication, which is generated by using the secret key shared between the source and destination nodes.

If the source and destination nodes do not share a secret key, an option is that the two ends first set up a route using the proposed routing protocol. Through this route the two nodes can establish a secret key with a two-party key exchange protocol [18]. Then end-to-end authentication can be implemented for subsequent messages based on the shared secret key. Certainly, the key exchange process will cause some overhead.

VI. Related Work

Yi et al. [29] developed a secure aware routing (SAR) protocol for ad hoc networks, which extended the Ad Hoc On-demand Distance Vector (AODV) routing protocol. In their protocol, the nodes in an ad hoc network have different security attributes and are classified into different trust levels. The trust level can be decided by an internal hierarchy of privileges in an organization. The nodes of the same trust level share a secret key. When a source constructs a route discovery message, it also specifies the required security level for the route. The route discovery message can also be encrypted by using the secret key shared by nodes of same trust level. Only the intermediate nodes that satisfy the required security level can process the message since only these nodes can decrypt the message. Other nodes just drop it. This protocol provides some protection to routing messages. The remaining problems are: Is the trust level fixed or can be changed? How to distribute key within the same trust level?

Papagiotis and Haas [30] proposed a secure routing protocol (SRP) for ad hoc networks. The assumption of SRP is the existence of a Security Association between a source node and a destination node, through which the source node and the destination node can authenticate each other. SRP is base on source routing. The source node broadcasts a route request to discover a route to the destination node. When an intermediate node receives the route request, it appends its identifier in the request packet and relays the request. So when the destination node receives the route request, a route has been set up and carried in the route request. The destination node generates a route reply containing the route and sends it back to the source node along the reverse of the route. The most important secure measure used in SRP is Message Authentication Code, which is calculated by using the shared secret key between the two ends. Both the unchanged fields of route request and the route reply are covered by a MAC so that modification and IP spoofing from non-colluding attackers can be prevented during the process of route discovery.

Venkatraman and Agrawal [31] proposed a protocol based on public key cryptography. They assume the existence of a governing authority for the distribution of public keys. A source node generates a route request and digitally signs it using its private key. When a destination node sends a route reply back to the source node, public key cryptography is used for pair-wise authentication to exclude malicious nodes. If a node does not know a forwarding node's public key, they have to exchange public keys first. This pair-wise authentication is done by challenge and response process. The purpose of this protocol is to prevent external attacks.

A different approach, authenticated routing for ad hoc networks (ARAN), was developed by Dahill et al. [32]. ARAN relies on public key cryptography for authentication. They assume that each node has a public/private key pair, and there exists a trusted certificate server to issue a certificate to each node. ARAN consists of two stages. The goal of stage one is for a source node to set up a route to a destination node. The source node broadcasts a route discovery packet, containing its certificate and digitally signed by using its private key. When a node receives the packet, it signs the packet using its private key and attaches to the packet its certificate and broadcasts the packet. Upon receiving the packet, a node verifies the

signature with the attached certificate. The node then removes the signature of the broadcasting node, signs the packet with its private key, attaches its certificate, and rebroadcasts the packet. Eventually the destination node receives the packet and validates the signatures of the source node and the forwarding node with their certificates. The destination node constructs a reply, signs it and unicasts the reply back to the source over the reverse path. When an intermediate node receives the reply, with the same process as the route discovery packet, it verifies the signature, replaces the signature with its signature and relays the packet. Finally the source node can receive the reply. The optional second stage is used to discover the shortest path between two ends. The source node broadcasts a Shortest Path Confirmation message, which contains the same information as the route discovery packet plus the certificate of the destination node. The route discovery part of the Shortest Path Confirmation message is signed by the source and encrypted using the destination node's public key. When an intermediate node receives the message, it signs the message, appends its certificate, encrypts the message using the destination node's public key, and rebroadcasts it. When the destination node receives the message, it can know the length of the path from the included cryptographic credentials of the intermediate nodes. Several protocols [33]–[36] based on public key cryptography have been proposed to protect routing protocols for wired network.

Several efficient signature schemes based on hashing chains [37] have been proposed to protect routing messages [38]–[41] and broadcast message [42]–[44] of wired network. Hu et al. [23], [24] and Zapata [45] adopted hashing chains to authenticate routing updates for ad hoc network situation. Perrig et al. [46] utilized hashing chain in securing sensor network.

Marti et al. [15] proposed a watchdog and pathrater scheme to improve the throughput of an ad hoc network in the presence of misbehaving node. Watchdog keeps track of misbehaving nodes. Pathrater avoids routing through those misbehaving nodes.

Yang et al. [47] extended AODV with a self-organized security approach. A token is utilized for authentication within the network, which is issued with a decentralized scheme [17], [48]. Only with a valid token, can a node participate in route discovery and data packet delivery. Their protocol does not assume the existence of centralized trusted severs and is suitable for ad hoc network situation.

Awerbuch et al. [49] proposed a fault detection scheme to detect malicious links on a route between a source and a destination. The scheme is based on acknowledgements from some probe nodes on the route, which are specified by the source node. If the number of acknowledgement loss exceeds a particular threshold, a faulty link is considered to exist in the route. Then a binary search can detect the faulty link.

VII. Conclusions and Future Work

In this paper, we proposed a secure routing protocol for ad hoc networks with a shared group key as the sole assumption. The key security measures in this protocol are distributed authentication and Message Authentication Code. We developed a Distributed Authentication Model, with which different nodes can authenticate each other. Integrity is ensured by Message Authentication Code, which is calculated by using the shared group key or pair-wise shared secret keys. A node establishes shared secret keys only with its trustworthy neighbors rather than all network nodes. The protocol can prevent or detect most of the attacks common to ad hoc routing protocols. The protocol is also able to exclude attackers timely and proactively. Moreover the protocol is capable of discovering multiple routes existed between two nodes and is also appropriate for dynamically changing network topology.

Trust value system and trust evaluation functions are important components of the Distributed Authentication Model. Currently there are no standards [11]. We plan to optimize them and improve their accuracy in our future work.

Our secure routing protocol can detect attacks, such as the modification of RREQ (Section V-B.2.a) and the modification of sequence number (Section V-B.2.c). In our future work we will study the measures to take when these attacks are detected.

Acknowledgment

We are grateful to anonymous reviewers whose valuable comments help us to improve the paper. This research was partially supported by NSF Grants IIS-0242384 and IIS-0324836.

References

[1] C. E. Perkins and E. Royer, "Ad-hoc on-demand distance vector routing," in *Proc. of the 2nd IEEE Workshop on Mobile Computing Systems and Applications*, February 1999, pp. 90–100.

[2] E. Royer and C. Toh, "A review of current routing protocols for ad-hoc mobile wireless networks," *IEEE Personal Communications*, April 1999.

[3] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, pp. 153–181, 1996.

[4] Z. J. Haas, "A new routing protocol for the reconfigurable wireless network," in *Proceeding of 1997 IEEE 6th International Conference on Universal Personal Communications Record: Bridging the Way to the 21st Century (ICUPC'97)*, October 1997, pp. 562–566.

[5] V. Park and M. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proc. of INFOCOM'97*, 1997.

[6] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the ACM SIGCOMM'94*. ACM Press, 1994, pp. 234–244.

[7] R. Ogier, F. Templin, and M. Lewis, "Topology Dissemination Based on Reverse-path Forwarding (TBRPF)," *Request for Comments RFC 3684, February, 2004*, February 2004.

[8] L. Zhou and Z. J. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, November/December 1999.

[9] S. Wu, F. Wang, and B. Vetter, "Secure routing protocols: Theory and practice," NC State Univ., Tech. Rep., April 1998.

[10] B. Schneier, *Applied Cryptography: Protocols, Algorithms, and Source Code in C. 2nd Edition*. John Wiley & Sons, Inc., 1996.

[11] H. Abdul-Rahman and S. Hailes, "A distributed trust model," in *New Security Paradigms Workshop*, 1997.

[12] B. Schneier, *Secrets and Lies: Digital Security in a Networked World. 1st Edition*. John Wiley & Sons, Inc., 2000.

[13] H. Krawczyk, M. Bellare, and R. Canetti, "HMAC: Keyed-hashing for message authentication," *Request for Comments RFC 2104*, February 1997.

[14] R. Ramanujan, A. Ahamad, J. Bonney, R. Hagelstrom, and K. Thurber, "Techniques for intrusion-resistant ad hoc routing algorithms (TIARA)," in *IEEE Military Communications Conference*, October 2000.

[15] S. Marti, T. Giuli, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of MOBICOM 2000*, August 2000.

[16] J.-P. Hubaux, L. Buttyan, and S. Capkun, "The quest for security in mobile ad hoc networks," in *MobiHoc '01: Proceedings of the 2nd ACM international symposium on Mobile ad hoc networking & computing*, New York, NY, USA, 2001, pp. 146–155.

[17] J. Kong, P. Zerfos, H. Luo, S. Lu, and L. Zhang, "Providing robust and ubiquitous security support for mobile ad-hoc networks," in *IEEE International Conference on Network Protocols (ICNP)*, November 2001.

[18] N. Asokan and P. Ginzboorg, "Key agreement in ad-hoc networks," *Computer Communications*, vol. 23, no. 17, November 2000.

[19] K. Fall and K. Varadhan, "ns notes and documentation," *available from http://www.isi.edu/nsnam/ns/*, 2003.

[20] "IEEE Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," *IEEE Std. 802.11-1997*, 1997.

[21] B. Tuch, "Development of waveLAN, an ISM band wireless LAN," *AT&T Technical Journal*, vol. 72, no. 4, pp. 27–37, July-Aug. 1993.

[22] J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu, and J. G. Jetcheva, "A performance comparison of multihop wireless ad hoc network routing protocols," in *Proc. IEEE/ACM MOBICOM'98*, October 1998, pp. 85–97.

[23] Y. Hu, D. B. Johnson, and A. Perrig, "SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks," in *4th IEEE Workshop on Mobile Computing Systems and Applications*, June 2002.

[24] Y. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *8th ACM International Conference on Mobile Computing and Networking (MobiCom 2002)*, September 2002.

[25] F. Stajano and R. J. Anderson, "The resurrecting duckling: Security issues for ad hoc wireless networks," in *7th Security Protocols Workshop, volume 1796 of Lecture Notes in Computer Science*, 1999, pp. 172–194,.

[26] F. Stajano, "The resurrecting duckling - what next?" in *Security Protocols - 8th International Workshop*, April 2001.

[27] D. Balfanz, D. Smetters, P. Stewart, and H. C. Wong, "Talking to strangers: Authentication in ad-hoc wireless networks," in *Network and Distributed System Security Symposium Conference Proceedings*, February 2002.

[28] M. Steiner, G. Tsudik, and M. Waidner, "Key agreement in dynamic peer groups," *IEEE Trans. Parallel Distrib. Syst.*, vol. 11, no. 8, pp. 769–780, 2000.

[29] S. Yi, P. Naldurg, and R. Kravets, "Security-aware ad hoc routing for wireless networks, Tech. Rep. UIUCDCS-R-2001-2241, August 2001.

[30] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, January 2002.

[31] L. Venkatraman and D. P. Agrawal, "Security scheme for routing in adhoc networks," in *Proceedings of the 13th International Conference on Wireless Communications*, July 2001, pp. 129–146.

[32] B. Dahill, B. Levine, C. Shields, and E. Royer, "Secure routing protocol for ad hoc networks," U Mass, Tech. Rep. UM-CS-2001-037, 2001.

[33] B. Smith and J. Garcia-Luna-Aceves, "Securing the border gateway routing protocol," in *Proceedings of Global Internet*, November 1996.

[34] B. Smith, S. Murthy, and J. Garcia-Luna-Aceves, "Securing distance-vector routing protocols," in *Proceedings of the Symposium on Network and Distributed System Security (SNDSS'97)*, February 1997, pp. 85–92.

[35] S. Murphy and M. Badger, "Digital signature protection of the OSPF routing protocol," in *Proceedings of the Symposium on Network and Distributed System Security (SNDSS'96)*, February 1996, pp. 93–102.

[36] R. Perlman, "Network layer protocols with byzantine robustness," *PhD thesis, MIT LCS TR-429*, October 1988.

[37] R. C. Merkle, "A digital signature based on a conventional encryption function," in *Advances in Cryptology-CRYPTO'87*, August 1987.

[38] S. Cheung, "An efficient message authentication scheme for link state routing," in *13th Annual Computer Security Applications Conference*, 1997.

[39] R. Hauser, T. Przygienda, and G. Tsudik, "Reducing the cost of security in link-state routing," in *Symposium on Network and Distributed System Security (SNDSS'97)*, February 1997, pp. 93–99.

[40] L. Reyzin and N. Reyzin, "Better than BiBa: Short one-time signatures with fast signing and verifying," in *Information Security and Privacy-7th Australasian Conference ACSIP*, 2002.

[41] K. Zhang, "Efficient protocols for signing routing messages," in *Proceedings of the 1998 Internet Society (ISOC) Symposium on Network and Distributed System Security*, March 1998.

[42] A. Perrig, "The BiBa one-time signature and broadcast authentication protocol," in *8th ACM Conference on Computer and Communication Security*, November 2001.

[43] A. Perrig, R. Canetti, J. Tygar, and D. Song, "Efficient and secure source authentication for multicast," in *Network and Distributed System Security Symposium, NDSS'01*, February 2001.

[44] A. Perrig, R. Canetti, D. Song, and J. Tygar, "Efficient authentication and signing of multicast streams over lossy channels," in *IEEE Symposium on Security and Privacy*, May 2000.

[45] M. G. Zapata, "Securing ad hoc routing protocols," in *Workshop on Wireless Security (WiSe'02)*, September 2002.

[46] A. Perrig, R. Szewczyk, V. Wen, D. Cullar, and J. D. Tygar, "SPINS: Security protocols for sensor networks," in *Proceedings of MOBICOM*, 2001.

[47] H. Yang, X. Meng, and S. Lu, "Self-organized network layer security in mobile ad hoc networks," in *Workshop on Wireless Security (WiSe'02)*, September 2002.

[48] H. Luo and S. Lu, "Ubiquitous and robust authentication services for ad hoc wireless networks," UCLA Computer Science, Tech. Rep. 200030, October 2000.

[49] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens, "An on-demand secure routing protocol resilent to byzantine failures," in *Workshop on Wireless Security (WiSe'02)*, September 2002.