I. J. Computer Network and Information Security, 2012, 12, 65-72
Published Online November 2012 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2012.12.07

Security Mechanisms to Decrease Vulnerability of Ad-hoc Routing Protocols

G.Sunayana, Sukrutharaj.M, Lalitha rani.N, M.B.Kamakshi
R.V. College of Engineering, Bangalore, Karnataka, India,
sunayana91@gmail.com, sukrutharajm@gmail.com, lalitharani13890@gmail.com, kamakshimb@gmail.com

Abstract — Many proposed routing protocols for ad hoc networks operate in an ad hoc fashion, as on demand routing protocols often have low overhead and faster reaction time than other types of routing based on periodic protocols. Dynamic nature of ad-hoc networks leads to challenges in securing the network. Due to the vulnerable nature of ad-hoc networks there are many security threats. One of the solutions to the problem is ARAN – Authenticated routing protocol which is a secure protocol and provides Integrity, Availability, Confidentiality, Authenticity, Non repudiation, Authorization & Anonymity. But an authenticated selfish node can interfere this protocol and disturb the network by dropping packets. However varieties of attacks targeting routing protocols have been identified. By attacking, the routing protocol attacker can absorb network traffic, inject them in the path between source and destination and thus control 0network traffic. Therefore many secure routing protocols have been developed that deal with these attacks. This paper analyzes the security aspects of one commonly used secure routing protocol ARAN.

Index terms — Routing protocols, ad hoc networks, Authentication, Non repudiation, Confidentiality, Authorization, ARAN

I. Introduction

A mobile ad hoc network (MANET), is a self-configuring infra structure less network of mobile devices connected by wireless links. Each device in a MANET is free to move independently in any direction, and will therefore change its links to other devices frequently. Each must forward traffic unrelated to its own use, and therefore be a router. The most common Routing protocol is Ad-hoc On Demand Distance Vector (AODV)[1] that handles the dynamically changing network well but only performs very basic security functions. With MANET being used for applications like on-line banking, business sensitive applications, and transfers of military information, security is much more important. From the viewpoint of security any routing protocol must satisfy the following criteria:

1) Identification of existence of nodes: If a route between two points in a network exists, it should always be possible to find it. Also, the node, which requested the route, should be able to be sure it has found a route to the correct node.

2) Identification of malicious nodes: The protocol should be able to identify misbehaving nodes and make them unable to interfere with routing. Alternatively, the routing protocol should be designed to be immune to malicious nodes.

3) Lightweight computations: Many devices connected to an ad-hoc network are assumed to be battery powered with limited computational abilities. Such a node cannot be expected to be able to carry out expensive computations. If operations such as public key cryptography or shortest path algorithms for large networks prove necessary, they should be confined to the least possible number of nodes; preferably only the route endpoints at route creation time.

4) Location privacy: Often, the information carried in message headers is just as valuable as the message itself. The routing protocol should protect information about the location of nodes in a network and the network structure.

5) Self-stabilization: The self-stabilization property requires that a routing protocol should be able to automatically recover from any problem in a finite amount of time without human intervention. That is, it must not be possible to permanently disable a network by injecting a small number of malformed packets. If the routing protocol is self-stabilizing, an attacker who wishes to inflict continuous damage must remain in the network and continue sending malicious data to the nodes, which makes the attacker easier to locate.

Securing protocols for mobile ad hoc networks presents unique challenges due to characteristics such as lack of pre-deployed infrastructure, centralized policy and control. We define and distinguish the heterogeneous environments that make use of ad hoc routing and differ in their assumed pre-deployment and security requirements. This approach is important because satisfying a tighter set of security requirements than an application requires is unwarranted and wasteful of resources.

We propose a secure routing protocol, Authenticated Routing for Ad hoc Networks (ARAN), that detects and protects against malicious actions by third parties and peers. ARAN introduces *authentication*, *message integrity*, and *non-repudiation* to routing in an ad hoc environment as a part of a minimal security policy, denial of-service attacks. Our proposed protocol, Authenticated Routing for Ad hoc Networks (ARAN), detects and protects against malicious actions by third parties and peers in one particular ad hoc environment.

The remaining sections of this paper are arranged as follows: Section II explains ad hoc networks and gives the major difference between ad hoc networks and IP networks. Section III explains the major drawbacks of protocols like AODV and DSR and also presents a table which compares AODV, DSR and ARAN. Section IV lists the requirements of a secure routing protocol and then Section V goes on to describe such a protocol called ARAN and its working. Section VI presents the conclusion drawn in this paper and the future work that can be carried out in this domain is explained in section VII.

II. BACKGROUND

An ad hoc network forms when a collection of mobile nodes join together and create a network by agreeing to route messages for each other. There is no shared infrastructure in an ad hoc network, such as centralized routers or defined administrative policy. All proposed protocols [2, 3, 4, 5, 6] have security vulnerabilities and exposures that easily allow for routing attacks. While these vulnerabilities are common to many protocols, in this paper we focus on two protocols that are under consideration by the IETF for standardization: AODV and DSR [6, 2].

The fundamental differences between ad hoc networks and standard IP networks necessitate the development of new security services. In particular, the measures proposed for IPSec [7] help only in end-to-end authentication and security between two network entities that already have routing between them; IPSec does not secure the routing protocol.

This point has been recognized by others. Zhou and Haas have proposed using threshold cryptography for providing security to the network [8]. Hubaux, et al. have proposed a method that is designed to ensure equal participation among members of the ad hoc group, and that gives each node the authority to issue certificates [9]. Kong, et al. [10] have proposed a secure ad hoc routing protocol based on secret sharing; unfortunately, this protocol is based on erroneous assumptions, e.g., that each node cannot impersonate the MAC address of multiple other nodes. Yi, et al. also have proposed a general framework for secure ad hoc routing [11].

III. DRAWBACKS OF EXISTING PROTOCOLS

The current proposed routing protocols for ad hoc wireless networks allow for many different types of attacks. Analogous exploits exist in wired networks [12], but are more easily defended against by infrastructure present in a wired network. In this section, we classify *modification*, *impersonation*, and *fabrication* exploits against ad hoc routing protocols.

Our focus is on vulnerabilities and exposures that result from the specification of the ad hoc routing protocol, and not from problems with IEEE 802.11 [13, 14, 15]. Additionally, trivial denial-of-service attacks based on interception and noncooperation are possible in all ad hoc routing protocols. While these attacks are possible, they are not achieved through subversion of the routing protocol.

The attacks presented below are described in terms of the AODV and DSR protocols, which we use as representatives of ad hoc on-demand protocols. Table 1 provides a summary of each protocol's vulnerability to the following exploits.

3.1Attacks Using Modification

Malicious nodes can cause redirection of network traffic and DoS attacks by altering control message fields or by forwarding routing messages with falsified values. For example, in the network illustrated in Fig. 1a, a malicious node M could keep traffic from reaching X by consistently advertising to B a shorter route to X than the route to X that C advertises. Below are detailed several of the attacks that can occur if particular fields of routing messages in specific routing protocols are altered or falsified.

3.1.1Redirection by modified route sequence numbers

Protocols such as AODV and DSDV [16] instantiate and maintain routes by assigning monotonically increasing sequence numbers to routes toward specific destinations. In AODV, any node may divert traffic through itself by advertising a route to a node with a *destination sequence num* greater than the authentic value. Fig. 1b illustrates an example ad hoc network. Suppose a malicious node, *M*, receives the RREQ that originated from *S* for destination *X* after it is re-broadcast by during route discovery. *M* redirects traffic toward itself by unicasting to *B* an RREP containing a much higher *destination sequence num* for *X* than the value last advertised by *X*.

Eventually, the RREQ broadcast by *B* will reach a node with a valid route to *X* and a valid RREP will be unicast back to- ward *S*. However, at that point *B* will have already received the false RREP from *M*.

If the *destination sequence num* for *X* that *M* used in the false RREP is higher than the *destination sequence num* for *X* in the valid RREP, *B* will drop the valid RREP, thinking that the valid route is stale. All subsequent traffic destined for X that travels through B will be directed toward M. The situation will not be corrected until either a legitimate RREQ or a legitimate RREP with a *destination sequence num* for X higher than that of M's false RREP enters the network.

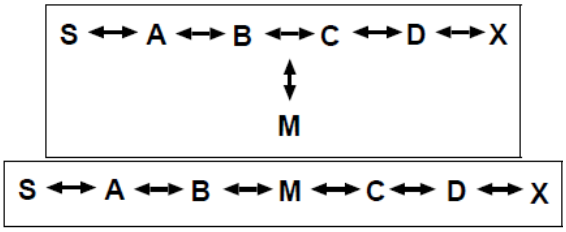

Figure 1a and 1b: Examples of two simple Adhoc Networks

Table 1: Vulnerabilities of AODV and DSR

Attack	AODV	DSR	ARAN
Remote redirection			
modif. of seq. numbers	Yes	No	No
modif. of hop counts	Yes	No	No
modif. of source routes	No	Yes	No
tunneling	Yes	Yes	Yes, but only to lengthen path
Spoofing	Yes	Yes	No
Fabrication			
fabr. of error messages	Yes	Yes	Yes, but non-repudiable
fabr. of source routes (cache poisoning)	No	Yes	No

3.1.2 Redirection with modified hop counts

A redirection attack is possible by modification of the hop count field in route discovery messages. When routing decisions cannot be made by other metrics, AODV uses the hop count field to determine a shortest path. In AODV, malicious nodes can increase the chances they are included on a newly created route by resetting the hop count field of the RREQ to zero. Similarly, by setting the hop count field of the RREQ to infinity, created routes will tend to not include the malicious node. Such an attack is most threatening when combined with spoofing, as detailed in Section 3.2.

3.1.3 Denial-of-service with modified source routes

DSR utilizes source routes, thereby explicitly stating routes in data packets. These routes lack any integrity checks and a simple denial-of-service attack can be launched in DSR by altering the source routes in packet headers.

Assume a shortest path exists from S to X as in Fig. 1b. Also assume that C and X cannot hear each other, that nodes B and C cannot hear each other, and that M is a malicious node attempting a denial-of-service attack. Suppose S wishes to communicate with X and that S has an unexpired route to X in its route cache. S transmits a data packet toward X, with the source route S->A->B->M->C->D->X contained in the packet's header. When M receives the packet, it can alter the source route in the packet's header, such as deleting from the source route. Consequently, when D receives the C altered packet, it attempts to forward the packet to X. Since X cannot hear C, the transmission is unsuccessful.

DSR provides a route maintenance mechanism such that a node forwarding a packet is responsible for confirming that the packet has been received by the next hop along the path. If no confirmation of receipt is received after retransmitting the packet a specified maximum number of attempts, this node should return a route error message to the source node.

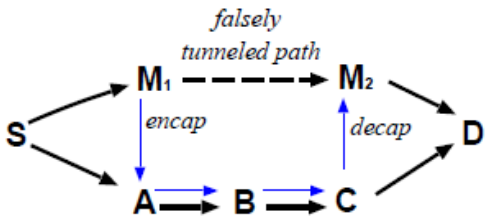

Figure 2 Path lengths spoofed by tunneling

In this case, C would send a route error message to S. Since M would be the first hop the route error takes on its path back to S, M can continue the denial-of-service attack by dropping this route error message.

DSR implements another route maintenance mechanism called *route salvaging* to recover from broken links along a path. When a break occurs, the node immediately upstream can check its route cache, and if it has a different route to that destination, it can use that route instead. In the example C would check its route cache for an alternate route. If C only knows of the erroneous route to X, the DoS attack can be completed.

Modifications to source routes in DSR may also include the introduction of loops in the specified path. Although DSR prevents looping during the route discovery process, there are insufficient safeguards to prevent the insertion of loops into a source route after a route has been salvaged[1].

3.1.4 Tunneling

Ad hoc networks have an implicit assumption that any node can be located adjacent to any other node. A *tunneling* attack is where two or more nodes may collaborate to encapsulate and exchange messages between them along existing data routes. One vulnerability is that two such nodes may collaborate to falsely represent the length of available paths by encapsulating and tunneling between them legitimate routing messages generated by other nodes. In this case, tunneling prevents honest intermediate nodes from correctly implementing the metric used to measure path lengths.

Fig. 2 illustrates such an attack where M_1 and M_2 are malicious nodes collaborating to misrepresent available path lengths by tunneling route request packets (e.g., an RREQ in AODV). Solid lines denote actual paths between nodes, the thin line denotes the tunnel, and the dotted line denotes the path that M_1 and M_2 falsely claim is between them. Node S wishes to form a route to D and initiates route discovery.

When M_1receives a RREQ from S, M_1 encapsulates the RREQ and tunnels it to M_2 through an existing data route, in this case M_1->A->B->C->M_2. When M_2 receives the encapsulated RREQ, it forwards the RREQ on to D as if it had only traveled S->M_1->M_2->D. Neither M_1 nor M_2 update the packet header to reflect that the RREQ also traveled the path A->B->C. After route discovery it appears to the destination that

there are two routes from S of unequal length: S->A->B->C->D; and S->M_1->M_2->D. If M_2 tunnels the RREP back to M_1, S would falsely consider the path to D via M_1 a better choice (in terms of path length) than the path to D via A .

Similarly, tunneling attacks are also a security threat to *multipath* routing protocols, which look for maximally disjoint paths [11]. In Fig. 2, two malicious nodes M_1 and M_2 may collaborate to tunnel routing messages to one another so that D falsely believes that the shortest route from S is S->M_1->M_2->D, as in the above attack. The paths S->A->B->C->D and S->M_1->M_2->D would appear completely disjoint, but actually share three common intermediate nodes, A, B, and C.

It is difficult to guarantee the integrity of path lengths with metrics like hop count. If route instantiation is determined by metrics that are governed solely by the operation of the routing protocol (such as a hop count metric), tunneling can cause routing metrics to be misrepresented.

Only an unalterable physical metric such as time delay can provide a dependable measure of path length. Specifically, a secure protocol must regard as the shortest path, the path that had the shortest delay of routing messages.

3.2 Attacks Using Impersonation

Spoofing occurs when a node misrepresents its identity in the network, such as by altering its MAC or IP address in outgoing packets, and is readily combined with modification attacks. The following example illustrates how an impersonation attack can work in AODV. Similar attacks are possible in DSR (see Table 1).

3.2.1 Forming Loops by Spoofing

Assume a path exists between the five nodes illustrated in Fig. 3a toward some remote destination, X, as would follow after an AODV RREQ/RREP exchange. In this example, A can hear B and D; B can hear A and C; D can hear A and C;

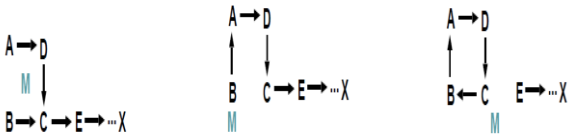

Figure 3. A sequence of events that form loops by spoofing of packets.

A malicious attacker, M, can learn this topology by listening to the RREQ/RREP exchanges during route discovery. M can then form a routing loop so that none of the four nodes can reach the destination. To start the attack, M changes its MAC address to match A's, moves closer to B and out of the range of A. It then sends an RREP to B that contains a hop count to X that is less than the one sent by C, e.g., zero. B therefore changes its route to the destination, X, to go through A, as illustrated in Fig. 3b. M then changes its MAC address to match B's, moves closer to C and out of range of B, and then sends to C an RREP with a hop-count to X lower than what was advertised by E. C then routes to X through B, as shown in Fig. 3c. At this point a loop is formed and X is unreachable from the four nodes. The attack is possible with a single malicious attacker; however, multiple attackers may collaborate for the same result.

3.3Attacks Using Fabrication

The generation of false routing messages can be classified as fabrication attacks. Such attacks can be difficult to verify as invalid constructs, especially in the case of fabricated error messages that claim a neighbor cannot be contacted.

3.3.1 Falsifying Route Errors in AODV and DSR

AODV and DSR implement path maintenance to recover broken paths when nodes move. If the source node moves and the route is still needed, route discovery is reinitiated with a new route request message. If the destination node or an intermediate node along an active path moves, the node upstream of the link break broadcasts a *route error* message to all active up- stream neighbors. The node also invalidates the route for this destination in its routing table[2].

The vulnerability is that routing attacks can be launched by sending false route error messages. Suppose node S has a route to node X via nodes A, B, C and D, as in Fig. 1. A malicious node can launch a denial-of-service attack against by continually sending route error messages to B spoofing node C, indicating a broken link between nodes C and X. B receives the spoofed route error message thinking that it came from C.

B deletes its routing table entry for X and forwards the route error message on to A, who then also deletes its routing table entry. If M listens and broadcasts spoofed route error messages whenever a route is established from S to X, M can successfully prevent communications between S and X.

3.3.2 Route Cache Poisoning in DSR

Corrupting routing state is a passive attack against routing integrity. This occurs when information stored in routing tables at routers is deleted, altered or injected with false information. Wired networks have been vulnerable to similar attacks [16, 19] but can often be defended against by security measures at routers.

Poisoning of route caches is a common example of this attack. The following details such an attack in DSR. In addition to learning routes from headers of packets that a node is processing along a path, routes in DSR may also be learned from promiscuously received packets. A node overhearing any packet may add the routing information contained in that packet's

header to its own route cache, even if that node is not on the path from source to destination. For example, in Fig. 1 a path exists from node S to node X via nodes A, B, C and D. If a packet traveling along the source route from S to X is overheard by another node, that node may then add the route <*S,A,B,C,D,X*> to its route cache.

The vulnerability is that an attacker could easily exploit this method of learning routes and poison route caches. Suppose a malicious node M wanted to poison routes to node X. If M were to broadcast spoofed packets with source routes to X via itself, neighboring nodes that overhear the packet transmission may add the route to their route cache. Since this route discovery feature of caching overheard routing information is optional in DSR, this exploit can be easily patched by disabling this feature in the network. The downside of this is that without this feature DSR operates at a loss in efficiency.

VI. SECURITY REQUIREMENTS OF AD HOC NETWORKS

A good secure routing algorithm prevents each of the exploits and it must ensure that no node can prevent successful route discovery and maintenance between any other nodes other than by non-participation. In sum, all secure ad hoc routing protocols must satisfy the following requirements to ensure that path discovery from source to destination functions correctly in the presence of malicious adversaries: (1) Route signaling cannot be spoofed; (2) Fabricated routing messages cannot be injected into the network; (3) Routing messages cannot be altered in transit, except according to the normal functionality of the routing protocol; (4) Routing loops cannot be formed through malicious action; (5) Routes cannot be redirected from the shortest path by malicious action.

The above requirements comprise the security needs of an *open* environment. The following additional requirement distinguishes a *managed open* environment: (6) Unauthorized nodes should be excluded from route computation and discovery. This requirement does not preclude the fact that authenticated peers may act maliciously as well. Additionally we assume that the managed-open environment has the opportunity for pre-deployment or exchange of public keys, session keys, or certificates.

We define a *managed hostile* environment to have requirements listed above as well as: (7)The network topology must not be exposed neither to adversaries nor to authorized nodes by the routing messages. Exposure of the net- work topology may be an advantage for adversaries trying to destroy or capture nodes.

V. SECURE AD HOC ROUTING PROTOCOL

AODV does not satisfy the requirements of certain discovery, isolation or Byzantine robustness. So secure routing protocol for ad hoc networks were developed, in order to offer protection against the attacks. These proposed solutions are either completely new stand-alone protocols, or in some cases incorporations of security mechanisms into existing protocols (e.g. DSR and AODV). A common design principle in all the proposals is the performance security trade-off balance. Since routing is an essential function of ad hoc networks, the integrated security procedures should not hinder its operation. Another important part of the analysis is the examination of the assumptions and the requirements on which each solution depends. Although a protocol might be able to satisfy certain security constraints, its operational requirements might thwart its successful employment.

Five most common categories of secure routing protocol are: solutions based on asymmetric cryptography; solutions based on symmetric cryptography; hybrid solutions; reputation-based solutions; and a category of mechanisms that provide security for ad hoc routing. In this paper one of most common and most efficient algorithm that is ARAN is chosen for analysis with respect of security from asymmetric cryptographic solution. This paper firstly presents a short description of ARAN then it briefly describes the analysis of ARAN in presence of above discussed attacks.

VI. ASYMMETRIC CRYPTOGRAPHIC SOLUTIONS

Protocols that use asymmetric cryptography to secure routing in mobile ad hoc networks require the existence of a universally trusted third party (TTP).

ARAN or authenticated routing protocol detects and protects against malicious actions by third party and peers in ad hoc network. Two distinct stages of ARAN consist of a preliminary certification process followed by a route instantiation process that guarantees end-to-end authentication. ARAN makes the use of cryptographic certificate to accomplish its task.

Route Initiation Step

Stage 1 each node, before attempting to connect to the ad hoc network, must contact the certification authority and request a certificate for its address and public key.

T→ A: cert A= [IPA, K_{A+} ,t, e]K_{T-} The certificate contains the IP address of A (IPA), the public key of A (K_{A+}), a timestamp k of when the certificate was created, and a time e at which the certificate expires. These variables are concatenated and signed by K_{T-}. The protocol assumes that each node knows a priori the public key of the certification authority.

Stage 2 The second operational stage of the protocol ensures that the intended destination was indeed reached. Each node must maintain a routing table with entries that correspond to the source-destination pairs that are currently active. The route discovery of the ARAN protocol begins with a node broadcasting a

route discovery packet (RDP) to its neighbors. A→ brdcst: [RDP, IPX, NA] KA-, $Cert_A$

The RDP includes a packet type identifier ("RDP"), the IP address of the destination X (IP_X), A 's certificate ($cert_A$) and a nonce N_A , all signed with A 's private key. Note that the RDP is only signed by the source and not encrypted, so the contents can be viewed publicly. The purpose of the nonce is to uniquely identify an RDP coming from a source. Each time, A, performs route discovery it monotonically increases the nonce.

Each node validates the signature with the certificate, updates its routing table with the neighbor from which it received the RDP, signs it, and forwards it to its neighbors after removing the certificate and the signature of the previous node (but not the initiator's signature and certificate).

Let B be a neighbor that has received from A the RDP broadcast, which it subsequently rebroadcasts.

B → brdcst: [[RDP, IP_X, N_A] K_{A-}] K_{B-}, $Cert_A$, $Cert_B$

Upon receiving the RDP B's neighbor C validates the signatures for both the RDP initiator, and B, the neighbor it received the RDP from, using the certificates in the RDP. C then removes B's certificate and signature, records as its predecessor, signs the contents of the message originally broadcast by Y and appends its own certificate C then rebroadcasts the RDP.

C→ brdcst: [[RDP, IP_X, N_A] K_{A-}] $K_{X\_}$, $Cert_A$, $Cert_C$. Eventually, the message is received by the destination X, who replies to the first RDP that it receives for a source and a given nonce. This RDP need not have traveled along the path with the least number of hops; the least-hop path may have a higher delay, either legitimately or maliciously manifested. In this case, however, a non-congested, non least hop path is likely to be preferred to a congested least hop path because of the reduction in delay. Because RDP's do not contain a hop count or specific recorded source route, and because messages are signed at each hop, malicious nodes have no opportunity to redirect traffic. After receiving the RDP, the destination unicasts a Reply (REP) packet back along the reverse path to the source. Let the first node that receives the REP sent by X be node D. X→ D: [REP, IP_A, N_A] K_{X-}, $cert_x$

The REP contains the address of the source node, the destination's certificate, a nonce, and the associated timestamp. The destination node signs the REP before transmitting it. The REP is forwarded back to the initiating node by a process similar to the process described for the route discovery, except that the REP is unicasted along the reverse path.

Let D's next hop to the source be node C .

D→ C : [[REP, IP_A, N_A] K_{X-}] K_{D-} , $cert_X$, $cert_D$

C validates D 's signature on the received message, removes the signature and certificate, then signs the contents of the message and appends its own certificate before unicasting the REP to B

C→ B : [[REP, IP_A, N_A] K_{X-}] K_{C-} ,$cert_x$, $cert_C$

Each node checks the nonce and signature of the previous hop as the REP is returned to the source. When the source receives the REP, it verifies the destination's signature and the nonce returned by the destination.

Route maintenance

When no traffic has occurred on an existing route for that route's lifetime, the route is simply de-activated in the route table. Data received on an inactive route causes nodes to generate an Error (ERR) message. Nodes also use ERR messages to report links in active routes that are broken due to node movement. All ERR messages must be signed. For a route between source A and destination X}, a node B generates the ERR message for its neighbor C as follows:

B → C: [ERR, IP_A, IP_X, N_b] K_{B-}, $cert_b$

This message is forwarded along the path toward the source without modification. A nonce ensures that the ERR message is fresh. It is extremely difficult to detect when

ERR messages are fabricated for links that are truly active and not broken. However, the signature on the message prevents impersonation and enables non-repudiation. A node that transmits a large number of ERR messages, whether the ERR messages are valid or fabricated, should be avoided

Key Revocation

In the event that a certificate needs to be revoked, the trusted certificate server, T, sends a broadcast message to the ad hoc group that announces the revocation. Calling the revoked certificate $cert_X$, the transmission appears as:

T → broadcast: [revoke, $cert_T$] K_{T-}.

Any node receiving this message re-broadcasts it to its neighbors. Revocation notices need to be stored until the revoked certificate would have expired normally. Any neighbor of the node with the revoked certificate needs to reform routing as necessary to avoid transmission through the now un trusted node.

Encryption

A -> B -> C -> D -> X

Decryption

A <- B <- C <- D <- X

VI. Conclusions

This paper has presented the authenticated routing protocol for securing the routing protocols of wireless networks. The study has demonstrated that inherent characteristics of ad hoc network such as lack of infrastructure network, rapidly changing topology adds difficulties to already complicated problem of secure routing [17]. Additionally, the flexibility of ad hoc networks enables them to be deployed in diverse application scenarios. Each application has its own set of security requirements and places unique demands on the underlying routing protocol. Hence, an additional difficulty in designing a secure protocol lies in the application scenario that is going to be protected and

how well the protocol can handle scenarios different than the scenario for which it has been designed. Authenticated routing protocol requires trusted third party for obtaining certificates. Therefore is preferable for applications where we can take help of some already existing infrastructure. ARAN protocol is based on Ad hoc on demand distance vector routing so as to take benefit of high performance and low cost due to it's on reactive nature. In this paper, we have introduced active attacks on AODV. This paper then discusses 5 types of active attacks. Generally, active attacks can be avoided by this use of stringer authentication methods. This paper firstly presents the complete working behind ARAN. As some limitations are also attached with every advantage, so is the case for ARAN. Apart from achieving so many security goals, it is also sufferer of weaknesses. For example ARAN does not have any mechanism that deals with black hole attack, wormhole attack, Denial of service attack.

VII. Future Scope

In this paper we identified different attacks on Authenticated Routing Protocol. ARAN has solution for some attacks but it is also silent about some attacks like black hole attack, denial of service attack etc. some research can be done to add functionality to ARAN that is also able to combat with above said attack. Areas in secure ad hoc network routing that have been explored are trust establishment [18, 19, 20, 21], key generation [22], nodes that maliciously do not forward packets [23], and security requirements for forwarding nodes [24]. These areas are beyond the scope of this paper. Routing protocol intrusion detection has been studied in wired networks as a mechanism for detecting misbehaving routers. Cheung and Levitt [25] and Bradley et al [26] propose intrusion detection techniques for detecting and identifying routers that send bogus routing update messages.

References

[1] Mobile Ad -hoc Networks (MANET). URL: http://www.ietf.org/html.charters/manet-charter.html.

[2] D. Johnson, D. Maltz, Y.-C. Hu, and J. Jetcheva. The dynamic source routing protocol for mobile ad hoc networks. *IEEE Internet Draft*, March 2001. draft-ietf-manet-dsr-05.txt (work in progress).

[3] S. Murthy and J.J. Garcia-Lunca-Aceves. An efficient routing protocol for wireless networks. *ACM Mobile Networks and Applications Journal*, pages 183–197, Oct. 1996.

[4] V. Park and M. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proc. INFOCOMM*, April 1997.

[5] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *Computer Communications Review*, pages 234–244, Oct. 1994.

[6] C. E. Perkins and E. M. Royer. Ad hoc on-demand distance vector rout- ing. In *IEEE Workshop on Mobile Computing Systems and Applications*, pages 90–100, Feb. 1999.

[7] C. R. Davis. *IPSec: Securing VPNs*. McGraw-Hill, New York, 2000.

[8] L. Zhou and Z. J. Haas. Securing ad hoc networks. *IEEE Network*, 13(6):24–30, 1999.

[9]J.P. HuBaux, L. Buttyan, and S. Capkun. The quest for security in mobile ad hoc networks. In *Proc. ACM MOBICOM*, Oct. 2001.

[10] J. Kong et al. Providing robust and ubiquitous security support for mobile ad-hoc networks. In *Proc. IEEE ICNP*, pages 251–260, 2001.

[11] S. Yi, P. Naldurg, and R. Kravets. Security-aware ad hoc routing for wireless networks. In *Proc. ACM Mobihoc*, 2001.

[12] F. Wang, B. Vetter, and S. Wu. Secure routing protocols: Theory and practice. Technical report, North Carolina State University, May 1997.

[13] W. Arbaugh, N.Shankar, and Y.C. Wan. Your 802.11 wireless network has no clothes. Technical report, Dept. of Computer Science, University of Maryland, March 2001.

[14] N. Borisov, I. Goldberg, and D.Wagner. Intercepting mobile communications: The insecurity of 802.11.http://www.isaac.cs.berkeley.edu/isaac/wep-faq.html.

[15] A. Stubblefield, J. Ioannidis, and A. D. Rubin. Using the fluhrer, mantin, and shamir attack to break wep. Technical Report TD-4ZCPZZ, AT&T Labs, August 2001.

[16] C. E. Perkins and P. Bhagwat. Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. *Computer Communications Review*, pages 234–244, Oct. 1994.

[17] E. M. Royer and C.-K. Toh, "A Review of Current Routing Protocols for Ad hoc Mobile Wireless Networks," IEEE Pers. Commun., vol. 2, no. 6, Apr. 1999, pp. 46–55.

[18] Dirk Balfanz, D. K. Smetters, Paul Stewart, and H. Chi Wong. Talking To Strangers: Authentication in Ad-Hoc Wireless Networks. In Symposium on Network and Distributed Systems Security (NDSS 2002), February 2002.

[19] Yih-Chun Hu, Adrian Perrig, and David B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In Proceedings of the Eighth Annual International Conference on Mobile Computing and Networking (MobiCom 2002), pages 12–23, September 2002.

[20] Jean-Pierre Hubaux, Levente Butty ´an, and Srdjan C ¡ apkun. The Quest for Security in Mobile Ad Hoc Networks. In Proceedings of the Third ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc 2001), Long Beach, CA, USA, October 2001

[21] Stefano Basagni, Kris Herrin, Emilia Rosti, and Danilo Bruschi. Secure Pebblenets. In ACM International Symposium on Mobile Ad Hoc Networking andComputing (MobiHoc 2001), pages 156–163, Long Beach, California, USA, October 2001.
[22] Sergio Marti, T.J. Giuli, Kevin Lai, and Mary Baker. Mitigating Routing Misbehaviour in Mobile Ad Hoc Networks. In Proceedings of the Sixth Annual InternationalConference on Mobile Computing and Networking (MobiCom 2000), pages 255–265, Boston MA, USA, August 2000.
[23] Seung Yi, Prasad Naldurg, and Robin Kravets. Security-Aware Ad- Hoc Routing for Wireless Networks. Technical Report UIUCDCS-R- 2001-2241, Department of Computer Science, University of Illinois at Urbana-Champaign, August 2001.
[24] Yih-Chun Hu, David B. Johnson, and Adrian Perrig. Secure Efficient Distance Vector Routing in MobileWireless Ad Hoc Networks. In Fourth IEEEWorkshop on Mobile Computing Systems and Applications (WMCSA '02), June 2002.
[25] Steven Cheung and Karl Levitt. Protecting Routing Infrastructures from Denial of Service Using Cooperative Intrusion Detection. In The 1997 New SecurityParadigms Workshop, September 1998.
[26] Kirk A. Bradley, Steven Cheung, Nick Puetza, Biswanath Mukherjee, and Ronald A. Olsson. Detecting Disruptive Routers: A Distributed Network Monitoring Approach. In Proceedings of the IEEE Symposium on Research in Security and Privacy, pages 115– 124, May 1998.

G.Sunayana has a B.E. in Telecommunication Engineering from R.V.College of Engineering, Bangalore and her interests are in the domain of Wireless and Telecommunication Networks and the different protocols used to optimise transmission of data.

Sukrutha Raj was a student of R.V. College of Engineering, Bangalore and obtained her B.E. in Telecommunication Engineering. Her domains of research include Computer Networks, Wireless Sensor Networks and Operating Systems.

Lalitha Rani has a B.E. in Telecommunication Engineering from R.V.College of Engineering, Bangalore. Her main areas of study are Very Large Scale Integrated circuits (VLSI), Computer Networks and Wireless Sensors.

Kamakshi M.B. is an assistant professor in the Telecommunication Department in R.V.College of Engineering, Bangalore. Her research is mainly in the areas of Computer Networks, Embedded Systems and Communication.

[1] There is also a potential for loops to form during route salvaging. An intermediate node salvaging the path replaces the source route in the packet with a new route from its route cache. DSR prevents infinite looping in this case by allowing a packet to only be salvaged a finite number of times.

[2] In DSR the source route is removed from the node's route cache.

