An Efficient Secure Zone Based Routing Protocol for Mobile Adhoc Network

Niroj Kumar Pani[1] and Dr. Ashok Kumar Turuk[2]

[1] Department of Computer Science and Engineering, National Institute of Technology
Rourkela-769008, India
E-mail: niroj_pani@yahoo.co.in, nirojpani@gmail.com

[2] Department of Computer Science and Engineering, National Institute of Technology
Rourkela-769008, India
E-mail: akturuk@nitrkl.ac.in, akturuk@gmail.com

Abstract: ***An ad hoc network is the cooperative engagement of a collection of mobile nodes without the required intervention of any centralized access point or existing infrastructure. There is an increasing trend to adopt ad hoc networking for commercial uses; however, their main applications lie in military, tactical and other security-sensitive operations. In these and other applications of ad hoc networking, secure routing is an important issue. Most of the secure routing protocols proposed in the literature are either proactive or reactive in nature. In this paper, we proposed a secure hybrid routing protocol for adhoc network, called Modified Secure Zone Routing Protocol (MSZRP), which is based on the concept of Zone routing protocol (ZRP). The paper details the design of the proposed protocol and analyses its robustness in the presence of multiple possible attacks that involves impersonation, modification, fabrication and replay of packets caused either by an external advisory or an internal compromised node. MSZRP successfully defeats all the identified threats. It is also resilient against the multilayer DoS attack***

Key words: *Ad-hoc Routing protocols, Security, Secure routing, Security attacks.*

1. Introduction

An ad hoc network is a collection of wireless computers (nodes), communicating among themselves over possibly multi-hop paths, without the help of any infrastructure such as base stations or access points [1, 2, 3, 4]. Unlike traditional mobile wireless networks, ad hoc networks have no fixed infrastructure. Mobile nodes that are within each other's radio range communicate directly via wireless links, while those far apart rely on other nodes to relay messages as routers. In ad hoc network each node acts both as a host (which is capable of sending and receiving) and a router which forwards the data intended for some other node. Applications of ad hoc network range from military operations and emergency disaster relief, to commercial uses such as community networking and interaction between attendees at a meeting or students during a lecture. Most of these applications demand a secure and reliable communication.

Ad hoc network routing protocols [5, 7, 8, 9, 10] are challenging to design and secure ones are even more, due to the unique characteristics of adhoc networks such as, lack of central authority, rapid node mobility, frequent topology changes, shared radio channel and limited availability of resources. A number of protocols have been proposed in the literature for secure routing. A survey of the protocols is given in [1, 2, 15]. Most of these protocols are either proactive or reactive in approach. However, both the approaches have their own limitations [10, 11]. For example, the proactive protocols use excess bandwidth in maintaining the routing information while, the reactive ones have long route request delay. Reactive routing also inefficiently floods the entire network for route determination.

In this paper, we proposed a secure hybrid ad hoc routing protocol, called Modified Secure Zone Routing Protocol (MSZRP), which takes the advantage of both proactive and reactive approach. Our proposed protocol is based on zone routing protocol (ZRP) [10, 11]. The reasons for selecting ZRP as the basis of our protocol are as follows: (i) ZRP is based on the concept of routing zones, a restricted area, and it is more feasible to apply the security mechanisms within a restricted area than in a broader area that of the whole network, (ii) Since the concept of zones separate the communicating nodes in terms of interior (nodes within the zone) and exterior (nodes outside the zone) nodes, certain information like network topology and neighbourhood information etc. can be hidden to the exterior nodes, (iii) Incase of a failure, it can be restricted within a zone.

We have also analyzed the robustness of MSZRP by evaluating it in the presence of the attacks introduced in [1] and [4]. The proposed protocol successfully detects and protects against all identified threats caused by both external and internal compromised nodes, by ensuring end to end authentication, message integrity and data confidentiality.

Rest of the paper is organized as follows: Overview of the protocol is presented in Section 2 and its architecture in Section 3. The routing algorithm is given in Section 4. Section 5 explains the process of proactive route computation and Section 6 the route maintenance services. The security aspect of the protocol is analyzed in Section 7. Finally some conclusions are drawn in Section 8.

2. Protocol Overview

The Modified Secure Zone Routing Protocol (MSZRP) is based on zone routing protocol (ZRP) [10, 11]. Like ZRP it performs intrazone [12] and interzone [13] routing; however, it differs from ZRP in security aspects. In ZRP where there is no security consideration, MSZRP is designed to address all measure security concerns like end to end authentication, message/packet integrity and data confidentiality during both intra and inter-zone routing. For end to end authentication and message integrity RSA digital signature mechanism [16] is employed, where as data confidentiality is ensured by an integrated approach of both symmetric and asymmetric key encryption [16]. Each communicating node has two pairs of private/public keys, one pair for signing and verifying and the other for encrypting and decrypting. For a node *X* the signing and verifying keys are SK_X and VK_X respectively while, encrypting and decrypting keys are EK_X and DK_X respectively. Among these keys SK_X and DK_X are private keys whereas VK_X and EK_X are public keys. Notations used in our proposed protocol are summarized in Table-1.

MSZRP makes the use of *public key certificates* [15, 16] for key distribution and management. Such certificates are already deployed as part of one-hop 802.11 networks [1]; this is the case on the UMass campus, where an 802.11 VPN is deployed and certificates are carried by nodes. For the process of public key certification, MSZRP assumes the presence of trusted certification servers called the *certification authorities* (CAs) in the network in addition to the communicating nodes which we call the *common nodes* (CNs). The public keys of the CAs are known to all valid CNs. Keys are generated apriori and exchanged through an existing, perhaps out of band, relationship between CA and each CN. Before entering the ad hoc network, each node requests a certificate from it's nearest CA. Each node receives exactly one certificate after securely authenticating their identity to the CA. The methods for secure authentication to the certificate server are numerous and hence it is left to the developers; a significant list is provided by [16]. A common node *X* receives a certificate from its nearest CA as follows:

$$CA \rightarrow X\text{: cert}_X = [IP_X, VK_X, EK_X, t, e] \mid sign_{CA}$$
$$\text{where, } sign_{CA} = [IP_X, VK_X, EK_X, t, e]\ SK_{CA}$$

The certificate contains the IP address of *X*, the two public keys VK_X and EK_X of *X*, one for verifying the signature signed by *X* and other for encrypting a packet to be send to *X*, a timestamp '*t*' of when the certificate was created, and a time '*e*' at which the certificate expires, all appended by the signature $sign_{CA}$ of CA. All nodes must maintain fresh certificates with their nearest CA.

Notations	Description
SK_X	Signature Key of node X (A private key used by X for signing)
VK_X	Signature verification key for node X. (A public key provided by X to verify its signature done with SK_X)
EK_X	Encryption Key for node X (A public key supplied by node X for encrypting any message to be sent to X)
DK_X	Decryption Key of node X (A private key used by X for decrypting any message which is encrypted with EK_X)
[d] SK_X	Packet 'd' signed with SK_X, this can be only verified using VK_X
{d}EK_X	Message 'd' encrypted with EK_X, this can be only decrypted with DK_X
[d] | b	b is appended to the packet containing d
$cert_X$	Public key certificate of X.
IP_X	IP address of X
t	Time stamp
e	Certificate expiration time
N_X	Nonce issued by node X
SKREQ	Session Key Request packet identifier
SKREP	Session Key Reply packet identifier
SRD	Secure Route Discovery packet identifier
SRR	Secure Route Reply packet identifier
ERR	Error packet identifier

Table 1: Notations Used

MSZRP is a two phase protocol. The first phase is the preliminary certification process where each CN fetches their required keys from their nearest CA. The second phase is secure routing phase which uses these keys to perform secure intra-zone or inter-zone routing.

3. MSZRP Architecture

The architecture of MSZRP is shown in Fig: 1. The proposed architecture is a modification of ZRP [10]. It is designed to support both secure routing (intrazone and interzone) and effective key management. There are dedicated and independent components in MSZRP to carry out these tasks. The functionality of each component and their interrelationship is explained below.

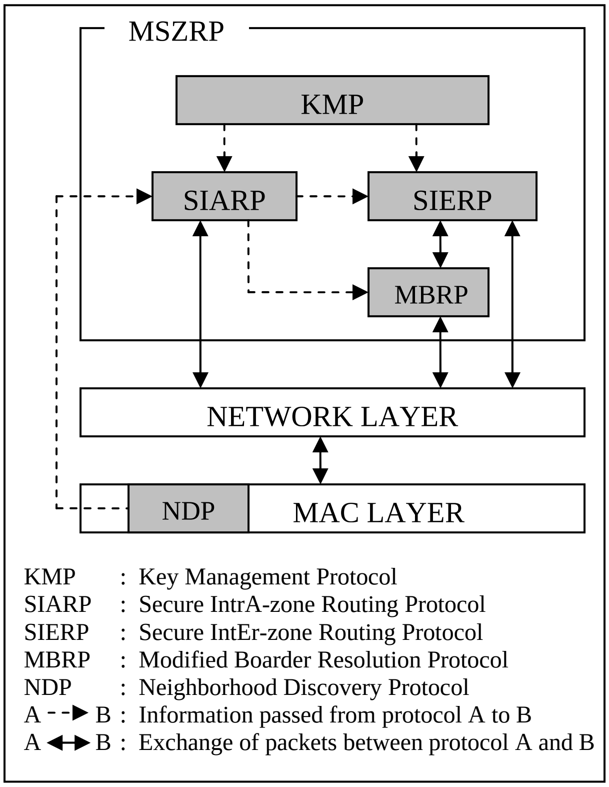
MSZRP
KMP
SIARP
SIERP
MBRP
NETWORK LAYER
NDP
MAC LAYER
KMP : Key Management Protocol
SIARP : Secure IntrA-zone Routing Protocol
SIERP : Secure IntEr-zone Routing Protocol
MBRP : Modified Boarder Resolution Protocol
NDP : Neighborhood Discovery Protocol
A - -▶ B : Information passed from protocol A to B
A ◀▶ B : Exchange of packets between protocol A and B

Fig 1: Architecture of MSZRP

The *key management protocol (KMP)* is responsible for public key certification process discussed in Section 2. It fetches the public keys for each CN by certifying them with the nearest CA. The *secure intrazone routing protocol (SIARP)* and *secure interzone routing protocol (SIERP)* uses these keys to perform secure intrazone and interzone routing respectively

SIARP is a limited depth proactive [5] link-state routing protocol [6] with inbuilt security features. It periodically computes the route to all intrazone nodes (nodes that are within the routing zone of a node) and maintains this information in a data structure called SIARP routing table. This process is called *proactive route computation*. The route information to all intrazone nodes collected in *proactive route computation* phase is used by SIARP to perform secure intrazone routing. Section 4.1 details secure intrazone routing and Section 5 proactive route computation.

SIERP is a family of reactive routing protocols [5] with added security features like ARAN [17]. It offers on demand secure route discovery and route maintenance services based on local connectivity information monitored by SIARP. The interzone routing and the route maintenance services offered by SIERP are discussed in Section 4.2 and Section 6 respectively

In order to detect the neighbor nodes and possible link failures, MSZRP relies on the *neighborhood discovery protocol (NDP)* [12] similar to that of ZRP. NDP does this by periodically transmitting a HELLO beckon (a small packet) to the neighbors at each node and updating the *neighbor table* [12] on receiving similar HELLO beckons from the neighbors. NDP gives the information about the neighbors to SIARP and also notifies SIARP when the *neighbor table* updates. We have assumed that NDP is implemented as a MAC layer protocol. A number of security mechanisms suggested in [4, 18] for MAC layer can be employed to secure NDP.

To minimize the delay during interzone route discovery, SIERP uses *bordercasting technique* [14] similar to ZRP, which is implemented here by the *modified border resolution protocol (MBRP)*. MBRP is a modification of the bordercast technique [14] adopted in ZRP. It not only forwards SIERP's secure route discovery packets to the peripheral nodes of the bordercasting node but also sets up a reverse path back to the neighbour by recording its IP address. MBRP uses the routing table of SIARP to guide these route queries. Since, all security measures are taken by SIERP during interzone routing; no additional security mechanism is adopted by MBRP during bordercasting.

4. Secure Routing

This Section describes the secure intrazone and interzone routing in details. We consider the network in Fig 2 for the illustration.

SIARP, at each node, periodically computes the route to all intrazone nodes and maintains this information in SIARP routing table. For example in Fig 2, node *A* proactively computes the route to *B, T, E, F* and *Y* and stores this information in its SIARP routing table. This process is called *proactive route computation*, discussed in Section 5.

When a node has a data packet for another node, it checks its SIARP routing table to determine whether the destination is within its zone or not. If the destination is within the zone, for example if node *A* has a packet destined for node *Y*, the packet is forwarded to the destination proactively using SIARP. On the other hand if the destination is outside the zone, for example if node *A* wants to transmit a packet to *Z,* then interzone routing is performed using SIERP

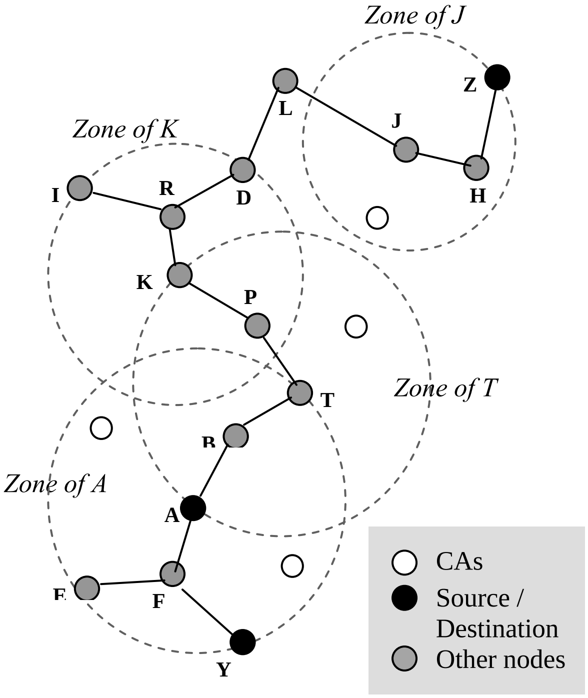

Fig 2: Intrazone and Interzone destinations of node A (zone radius $\beta = 2$).

4.1. Secure intrazone routing

For intrazone routing we consider *A* as the source and *Y* as the destination. The following steps are taken by SIARP (at node *A*) to route the data packet from *A* to *Y*.

Step 1: *A* looks for the route to *Y* in its SIARP routing table and finds it to be A-F-Y.

Step 2: *A* sends a SKREQ packet to *Y* along this route requesting a session key K_{AY} between *A* and *Y*.

$$A \rightarrow Y : [SKREQ, IP_Y, cert_A] \mid sign_A$$
$$\text{where, } sign_A = [SKREQ, IP_Y, cert_A]\ SK_A$$

The SKREQ packet contains a packet type identifier "SKREQ", the IP address of the destination *Y*, and *A'*s certificate, all appended by the signature $sign_A$ of *A* signed using SK_A.

Step 3: *Y* on receiving this request, verifies the signature using VK_A, which it extracts from *A*'s certificate, creates the session key K_{AY}, encrypts it using EK_A and sends it to *A* as SKREP packet along the reverse route Y-F-A.

$$Y \rightarrow A : [SKREP, IP_A, cert_Y, \{K_{AY}\}EK_A] \mid sign_Y$$
$$\text{where, } sign_Y = [SKREP, IP_A, cert_Y, \{K_{AY}\}EK_A]\ SK_Y$$

The packet contains a packet type identifier "SKREP", the IP address of *A*, the certificate of *Y* and the session key encrypted using EK_A, all appended by the signature $sign_Y$ of *Y* signed using SK_Y.

Step 4: *A* on receiving the SKREP packet, verifies it using VK_Y, conforms the authenticity of the packet, decrypts it using DK_A and extracts the session key K_{AY}.

Once *A* gets the session key, it can encrypt the data packet using K_{AY} and send it to *Y* along the same route A-F-Y. All further communication between *A* and *Y* takes place similarly, using this session key.

4.2. Secure interzone routing

Secure interzone routing is done using SIERP. The interzone routing is initiated with an on demand *secure route discovery* phase in which the source finds the route to the desired interzone destination. The source then sends the data packet along this route. In our case when *A* wants to send a packet to *Z*, *A* looks in its SIARP routing table for a valid route to *Z*. Since *Z* is not within the zone of *A*, *A* fails to find the route. In this case, *A* begins the secure route discovery process to *Z* The *secure route discovery process* gives *A* the authentic route to *Z* after which *A* forwards the data packet to *Z* along this route. In addition to secure route discovery, SIERP also performs route maintenance services based on the local connectivity information monitored by SIARP. Route maintenance is discussed in Section 6.

The following steps are taken by SIERP to route the data packet from *A* to *Z*:

Step 1: SIERP at *A* begins the *secure route discovery* process to *Z* by bordercasting to its peripheral nodes *T, E* and *Y,* a *SRD packet* with the help of MBRP.

$$A \rightarrow bordercast : [SRD, IP_Z, cert_A, \beta, N_A, t] \mid sign_A$$
$$\text{where, } sign_A = [SRD, IP_Z, cert_A, \beta, N_A, t]\ SK_A$$

The packet contains a packet type identifier "SRD", the IP address of the destination *Z*, A's certificate, the zone radius 'β', a nonce N_A created by *A* and the current time *t*, all appended by the signature $sign_A$ of *A*. The nonce N_A is monotonically increased every time *A* performs route discovery. N_A and *t* together with the IP address of *A* (IP_A) uniquely identify the SRD which prevents the replay attack. N_A is made large enough such that, it will not need to be recycled within the probable clock skew between receivers. If a nonce later reappears in a valid packet that has a later timestamp, the nonce is assumed to have wrapped around, and is therefore accepted. Note that a hop count is not included with the message.

Step 2: When a peripheral node of A (T, E *or* Y), receives the SRD, it checks the (IP_A, N_A, t) tuple to verify that it has not already processed this SRD. Nodes do process packets for which they have already seen this tuple. The receiving node uses A's public key, which it extracts from A's certificate, to validate the signature and verify that A's certificate has not expired. If the packet is found to be authentic, it sets up a reverse path back to the source A by recording the neighbor from which it received the SRD, for example when the peripheral node T receives the SRD it sets up a reverse path back to A by recording the neighbor B from which it received the SRD (B sets up a reverse path to A during bordercasting. Now, T sets up the reverse path to B. So a reverse path from T to A is set).

The peripheral node then signs the contents of the message originally bordercast by A and appends this signature and its own certificate to the SRD. It checks in its SIARP routing table whether it has a valid path to the destination Z. If it has (Z is within the zone of the node), it forwards the SRD directly to Z along this route, otherwise it rebordercasts the packet to its peripheral nodes. In the present case since none of the peripheral nodes T, E and Y has the route to Z (Z is not within the zone of T, E or Y), all rebordercasts the SRD to their peripheral nodes, for example, T rebordercasts the SRD to K.

$$T \rightarrow bordercast : [[SRD, IP_Z, cert_A, \beta, N_A, t] \mid sign_A] \mid sign_T, cert_T$$
$$\text{where, } sign_T = [[SRD, IP_Z, cert_A, \beta, N_A, t] \mid sign_A]]SK_T$$

Step 3: Upon receiving the SRD, T's peripheral node K checks the (IP_A, N_A, t) tuple, validates T's signature and sets up the reverse path to T (if the signature is authentic). K then removes T's certificate and signature, signs the contents of the message originally bordercast by A and appends this sign along with its own certificate to the SRD. It checks in its SIARP routing table whether it has a valid path to Z. Since it doesn't, it again rebordercasts the packet to its peripheral nodes I and D.

$$K \rightarrow bordercast : [[SRD, IP_Z, cert_A, \beta, N_A, t] \mid sign_A] \mid sign_K, cert_K$$
$$\text{where, } sign_K = [[SRD, IP_Z, cert_A, \beta, N_A, t] \mid sign_A]]SK_K$$

Each node along the path repeats these steps of validating the previous node's signature, recording the previous node's IP address for setting up the reverse path, removing the previous node's certificate and signature, signing the original contents of the message, appending its own certificate and rebordercasting the message, until the SRD reaches a node, that has a valid route to the destination Z (Z is within the zone of the node). In this case the node instead of rebordercasting the SRD, directly forwards it to Z. For example, when the SDR reaches J, it validates the packet, sets up the reverse path to the bordercasting node D, removes D's certificate and signature, signs the contents of the message originally bordercast by A, appends this signature and its certificate and forwards the SRD to Z.

$$J \rightarrow Z : [[SRD, IP_Z, cert_A, \beta, N_A, t] \mid sign_A] \mid sign_J, cert_J$$
$$\text{where, } sign_H = [[SRD, IP_Z, cert_A, \beta, N_A, t] \mid sign_A]]SK_J$$

Step 4: Finally, the SRD arrives at destination Z, which replies to the first SRD that it receives for a source and a given nonce. There is no guarantee that the first SRD received traveled along the shortest path from the source. A SRD that travels along the shortest path may be prevented from reaching the destination first if it encounters congestion or network delay, either legitimately or maliciously manifested. In this case, however, a non-congested, non-shortest path is likely to be preferred to a congested shortest path because of the reduction in delay. Because SRDs 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.

Z on getting this SRD packet verifies it using both VK_J and VK_A, confirms its authenticity and extracts EK_A. Z creates a *secure route reply* (SRR) packet and *unicasts* it back to the source along the reverse path. The first node that receives the SRR sent by Z is H.

$$Z \rightarrow H : [SRR, IP_A, cert_Z, N_A, t, \{K_{AZ}\}EK_A] \mid sign_Z$$
$$\text{where, } sign_Z = [SRR, IP_A, cert_Z, N_A, t, \{K_{AZ}\}EK_A]SK_Z$$

The SRR includes a packet type identifier "SRR", the IP address of A, the certificate of Z, the nonce N_A, the associated time stamp t sent by A and a session key K_{AZ} between A and Z encrypted with EK_A, all appended by the signature $sign_Z$ of Z. Nodes that receive the SRR forward the packet back to the predecessor from which they received the original SRD. Each node along the reverse path back to the source signs the SRR and appends its own certificate before forwarding the SRR to the next hop. Since, J is the next hop node to the source A after H:

$$H \rightarrow J : [[SRR, IP_A, cert_Z, N_A, t, \{K_{AZ}\}EK_A] \mid sign_Z] \mid sign_H, cert_H$$
$$\text{Where, } sign_H = [[SRR, IP_A, cert_Z, N_A, t, \{K_{AZ}\}EK_A] \mid sign_Z] SK_{H,}$$

J on getting the SRR validates *H*'s signature on it, removes *H*'s signature and certificate, signs the contents of the message and appends this signature and its own certificate before unicasting the SRR to its neighbour *L*.

.

$$J \rightarrow L : [[SRR, IP_A, cert_Z, N_A, t, \{K_{AZ}\}EK_A] \mid sign_Z] \mid sign_J, cert_J$$

$$\text{Where, } sign_J = [[SRR, IP_A, cert_Z, N_A, t, \{K_{AZ}\}EK_A] \mid sign_Z] SK_J$$

Each node checks the nonce and signature of the previous hop as the SRR is returned to the source. This avoids attacks involving impersonation and replay of the message. Eventually the source *A* receives the SRR.

Step 5: On getting the SRR, *A* verifies *Z*'s signature and the nonce returned by *Z* to conform its authenticity. It then extracts the session key K_{AZ}. *A* now encrypt the data packet using K_{AZ} and send it to *Z* along the same route.

5. Proactive Route Computation

For *proactive route computation* each node within its routing zone periodically advertises a *link state packet (LSP)*. For example, node *A* advertises the LSP within the zone of *A*.

$$A \rightarrow brdcast : [LSP, IP_A, cert_A, \beta, TTL, SNo, neighbour[n], link\_metric[n]] \mid sign_A$$

$$\text{Where, } sign_A = [LSP, IP_A, cert_A, \beta, TTL, SNo, neighbour[n], link\_metric[n]] SK_A$$

The packet contains a packet type identifier "LSP", the IP address of the broadcasting node *A*, the certificate of *A*, the zone radius '*β*', a time-to-live (TTL) value, the sequence number *SNo* of the packet which is used to track the link state history of the source node *A,* the list of neighbours of *A*, and link metrics, all appended by the signature $sign_A$ of *A*. The TTL field is used to control the scope of the packet which is initialized to *β-1* hops by *A*. Upon receipt the packet, the TTL value is decremented and as long as the value is greater than 0, the LSP is rebroadcasted.

When a neighbour of *A*, receives the LSP, it verifies the authenticity of the packet using VK_A which it extract from *A*'s certificate in the LSP, add LSP's information to its *link-state table* [6], decrement the value of TTL field and again forwards this LSP as long as the value of TTL field is greater than 0 else the LSP is dropped. Because every node within the zone of *A* receives the same LSPs, all the nodes build the same link state table. A typical link state table contains at least the following fields: < Source address, Zone radius, Neighbour ID, Insert time, route metrics >.

Once the link-state table is built, each node computes the route to every other node within its zone by applying the Dijkstra algorithm [6] to its link state table and stores this information in its SIARP routing table. A typical SIARP routing table maintained at a node contains the following fields: <Dest_Address, Routes, Route metrics> and has entries for all intrazone nodes.

6. Route Maintenance

MSZRP is a hybrid routing protocol. SIARP is proactive and SIERP is reactive in nature. SIARP doesn't mandate for route maintenance, as the node mobility within a zone is periodically updated. However, route maintenance is required in SIERP for interzone routing.

For route maintenance, SIERP at each node keeps track of routes whether they are active or not. When there is no flow of traffic on an existing route for that route's lifetime, the route is deactivated by the node. Data received on an inactive route causes nodes to generate an Error (ERR) message. A node generates an ERR message in either of the following cases: (i) if data is received on an inactive route, or (ii) the link of an active route is broken due to node mobility or some other reasons. The node send the ERR message to the source along the reverse path. All ERR messages must be signed to check the authenticity of the sender as well as the message. For a route between source *A* and destination *X*, a node *M* generates the ERR message for its neighbor *N* as follows:

$$M \rightarrow N : [ERR, IP_A, IP_X, cert_M, N_M, t] \mid sign_M$$

$$\text{Where, } sign_M = [ERR, IP_A, IP_X, cert_M, N_M, t] SK_M$$

This message is forwarded along the path to the source without modification. A nonce and timestamp ensure that the ERR message is a fresh. Since the ERR messages are signed, malicious nodes cannot generate ERR messages for other nodes. The non-repudiation provided by the signed ERR message allows a node to be verified as the source of each ERR message that it sends. The source node drops the duplicate ERR message with same nonce and time stamp.

7. Security Analysis of MSZRP

In this Section, we analyze the security aspects of MSZRP by evaluating its robustness in the presence of attacks mentioned in [1] and [4]. MSZRP can prevent attacks that include impersonation, modification, fabrication and replay of packets caused by both an external advisory and an internal compromised node within the network. It is also resilient against the multilayer denial of service attack.

Attacks involving impersonation: MSZRP participants, accept only those packets that have been signed with a certified key issued by a CA. In intrazone routing since the SKREQs and SKREPs can only be signed by an authenticated source with its own private signature key, nodes can't impersonate (spoof) other nodes. Interzone routing follows hop-by-hop authentication during route discovery and end-to-end authentication during the route reply phase. So it is impossible for an external node or an internal compromised node to impersonate an intermediate node during interzone routing. Further since the SRD packet is signed by the source node using its private key, it guarantees that only the source can initiate a route discovery process. Similarly, the SRR packets include the destination's certificate and signature, ensuring that only the destination can respond to the route discovery. This prevents attacks where the source, the destination or any intermediate nodes are spoofed e.g. blackhole, wormhole or DoS attacks.

Prevention from Information Disclosure: No hop count information is present in the SRD or SRR packets. This prevents an external advisory or an internal compromised node from getting any kind of information about the network topology. Topology information is restricted to nodes within a zone. This is harmless as nodes accept packets only after verifying the sender's signature. Further all the data packets and the control packets that contain the session key are encrypted which ensures the confidentiality.

Routing message Modification: MSZRP specifies that all fields of LSPs, SKREQ, SKREP, SRD and SRR packets remain unchanged between the source and the destination. Since all packets are signed by the initiating node, any alterations in transit would be immediately detected by intermediary nodes along the path, and the altered packet would be subsequently discarded. Repeated instances of altering packets could cause other nodes to exclude the errant node from routing. Thus, modification attacks like routing table poisoning are prevented.

Fabrication of routing messages: Messages can be fabricated only by the internal compromised nodes with certificates. In that case, MSZRP does not prevent fabrication of routing messages, but it does offer a deterrent by ensuring non-repudiation. A node that continues to inject false messages into the network may be excluded from future route computation.

Replay Attacks: Replay attacks like wormhole attack are prevented by including a nonce and a timestamp with routing messages.

8. Conclusion

In this paper, we have presented the design and analysis of a new secure ad hoc network routing protocol called Modified Secure Zone Routing Protocol (MSZRP). The proposed protocol is based on the concept of zone routing protocol (ZRP). In designing MSZRP, we carefully fit the inexpensive cryptographic primitives to each part of the protocol functionality to create an efficient protocol that is robust against multiple attacks in the network. MSZRP gives a better solution towards achieving the security goals like message integrity, data confidentiality and authentication, by taking an integrated approach of digital signature and both the symmetric and asymmetric key encryption technique. The proposed protocol intends to provide security at IP layer. Together with existing approaches for securing the physical layer and MAC layer within the network protocol stack, the Modified Secure Zone Routing Protocol (MSZRP) provides a foundation for the secure operation of an ad hoc network.

References

[1] C. Siva Ram Murthy and B. S Manoj, "Ad Hoc Wireless Networks, Architecture and Protocols", Prentice Hall PTR, 2004.

[2] Stefano Basagni, Macro Conti, Silvia Giordano and Ivan Stojmenovic, "Mobile Ad Hoc Networks" , IEEE press, A john Wily & Sons, INC. publication, 2003

[3] George Aggelou, "Mobile Ad Hoc Networks", 2nd edition, Mc GRAW Hill professional engineering, 2004

[4] Imrich Chlamtac, Marco Conti, Jenifer J.-N. Liu, "Mobile Ad Hoc Networking: Imperatives and Challanges", Elsevier Network Magazine, vol. 13, pages 13-64, 2003

[5] E.M. Belding-Royer and C.-K. Toh. "A review of current routing protocols for ad-hoc mobile wireless networks", IEEE Personal Communications Magazine, pages 46–55, April 1999.

[6] Behrouz A. Forouzan, "Data communication and Networking," 2nd edition, Tata McHill publication, 2001

[7] D.B. Johnson, D.A. Maltz, "Dynamic source routing in adhoc wireless networks", in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publishers, Dordrecht, 1996, pp. 153–181.

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

[9] P. Jacquet, P. Muhlethaler, A. Qayyum, "Optimized Link State Routing Protocol", Internet Draft, draft-ietf-manetolsr- 00.txt, November 1998.

[10] Haas Z. J., Pearlman M. R., and Samar P., "The Zone Routing Protocol (ZRP)", IETF Internet Draft, draft-ietf-manet-zone-zrp-04.txt, July 2002.

[11] Jan Schaumann, "Analysis of Zone Routing Protocol", Course CS765, Stevens Institute of Technology Hoboken, New Jersey, USA, 8th December 2002

[12] Haas, Zygmunt J., Pearlman, Marc R., Samar, P.: "Intrazone Routing Protocol (IARP)", IETF Internet Draft, draft-ietf-manet-iarp-01.txt, June 2001

[13] Haas, Zygmunt J., Pearlman, Marc R., Samar, P.: "Interzone Routing Protocol (IERP)", IETF Internet Draft, draft-ietf-manet-ierp-01.txt, June 2001

[14] Haas, Zygmunt J., Pearlman, Marc R., Samar, P.: "The Bordercast Resolution Protocol (BRP) for Ad Hoc Networks", IETF Internet Draft, draft-ietf-manet-brp-01.txt, June 2001

[15] L. Zhou and Z. J Haas, "Securing Ad Hoc networks," IEEE Network Magazine, vol. 13, no. 6, December 1999

[16] Behrouz A. Forouzan, "Cryptography and Network Security", Special Indian Edition, Tata McHill publication, 2007

[17] K.Sanzgir, and B.Dahill, "A secure routing protocol for ad hoc networks", Proceeding of the 10th IEEE International Conference on Network Protocols, 2002, pp.1-10.

[18] P. Michiardi, R. Molva, "Ad hoc networks security", in: S. Basagni, M. Conti, S. Giordano, I. Stojmenovic (Eds.), Ad Hoc Networking, IEEE Press Wiley, New York, 2003.

Biographies

Niroj Kumar Pani, received M.C.A (2004) from KIIT university and M.Tech (2009) in Computer Science with specialization in Information security from National Institute of Technology, Rourkela. He stated his carrier as a lecturer (2004-07) in Indian Institute of Science and Information Technology and at present he is working as a senior software analyst in PMAP India. His research interests include, network security, mobile communications and Ad hoc networks.

Dr. Ashok Kumar Turuk, received B.E (1992), M.E (2000) both from National Institute of Technology, Rourkela and Ph.D (2005) from IIT Kharagpur. He has published over twenty research papers in national and international journals/conferences field of Wireless & Digital Communication Network, and supervised more than 30 projects/dissertation of Ph.D, M.Tech. & B.Tech. students. Presently he is working as Asst. Professor in National Institute of technology, Rourkela in the department of Computer Science and Engineering.