This full text paper was peer reviewed at the direction of IEEE Communications Society subject matter experts for publication in the WCNC 2008 proceedings.

RSRP: A Robust Secure Routing Protocol for Mobile Ad hoc Networks

Syed Rehan Afzal, Subir Biswas, Jong-bin Koh, Taqi Raza, Gunhee Lee, and Dong-kyoo Kim
Graduate School of Information and Communication, Ajou University,
San 5 Wonchun Dong, Yeongtong, Suwon- 443 749, Republic of Korea
{rehan, subir, nitefly, taqi, icezzoco, dkkim}@ajou.ac.kr

Abstract— **Routing scenario in ad hoc networks is different from infrastructure-based wireless networks; since in ad hoc networks each node acts as a router and is responsible for managing topological information and ensuring correct route learning. Although a number of secure routing protocols have been proposed so far, all of them have certain advantages and disadvantages. Hence, security in ad hoc networks is still a contentious area. In this paper we first explore the security problems and attacks in existing routing protocols and then we present the design and analysis of a new secure on-demand routing protocol, called RSRP which confiscates the problems mentioned in the existing protocols. Moreover, unlike Ariadne, RSRP uses a very efficient broadcast authentication mechanism which does not require any clock synchronization and facilitates instant authentication.**

Keywords- Ad hoc network routing, security, networking

I. INTRODUCTION

Routing protocols in general can be classified into two categories: Proactive and Reactive routing protocols. In proactive routing protocols like Optimized Link State Routing (OLSR), Topology Broadcast based on Reverse Path Forwarding (TBRPF) and Hazy Sighted Link State Routing, routes are established before communication requirement and therefore the latency delays experienced while discovering new routes is avoided. On the contrary, in reactive routing protocols such as Ad hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR), route information is collected only at the time of communication requirement, hence conserving precious node battery. For the rest of the paper we'll basically focus on the reactive routing protocols; since in MANET environment, in most of the situations, they perform efficiently and incur low overhead as compared to the proactive routing protocols. Moreover, efficiency of a routing protocol is strongly dependent on the innocence of the participating nodes; nevertheless, the possibility of existence of malicious nodes is undeniable. Therefore, design of an efficient and secure routing protocol is a challenging issue.

In this paper, we make two contributions to the area of security in routing protocols for mobile ad hoc networks. First, we describe the problems in the existing routing protocols in terms of security and efficiency. Second, we present the design and security analysis of our proposed secure routing protocol RSRP, that copes with the problems mentioned in existing routing protocols.

In the sequel, Section 2 describes problems in existing protocols. In Section 3 of this paper the related work is mentioned, which comprises of a broadcast authentication scheme PARM [4]. We explain our proposed scheme RSRP in Section 4, followed by detailed protocol description in Section 5. In Section 6, we state the protocol analysis and finally in Section 7, we present our conclusions.

II. PROBLEMS IN EXISTING ROUTING PROTOCOLS

A number of routing protocols have been proposed so far. All of them have certain advantages and disadvantages. Unfortunately, prerequisite for all the available routing protocols is a managed environment characterized by some security infrastructure established prior to the security protocol execution. Since it is impossible to design a secure ad hoc routing protocol exclusive of any such assumption, we have tried not to point out any such assumptions unless they are too much unrealistic and impractical. In this section, we briefly describe problems in the existing routing protocols in MANETs emphasizing more on DSR and Ariadne, as our proposed scheme RSRP is also based on DSR.

A. DSR

As DSR is not a secure routing protocol, malicious node can modify source routes, forge route error messages, RREQ, RREP etc., or perform replay or tunneling attacks or perform route cache poisoning.

B. Ariadne

Ariadne is a secure routing protocol based on DSR. Ariadne can authenticate routing messages using one of three schemes: shared secrets between each pair of nodes, shared secrets between communicating nodes combined with broadcast authentication TESLA [2], or digital signatures. In Ariadne, source node S creates an encrypted request RREQ, and broadcasts message *{RREQ, h(S, D, id, time interval)}*, where h is the MAC computed using the MAC key K_{SD}. As soon as Node A receives this broadcast request from S, it broadcasts message *{RREQ, h^2(r), (A), (M_A)}*, where $h^i(r) = h[i,h^{i-1}(r)]$ having 'i' as the identity of the node performing the hash. For example in this case A computes $h^2(r)$ as $h[A, h^1(r)]$. M_A is the MAC computed on the entire request message including the new hash using the Tesla key of Node A. Note, the Tesla key of Node A will not be revealed to other nodes, until the RREQ sender i.e. S receives the reply message RREP in the worst possible case *(with maximum delay or from the longest possible path)*. Similarly, B receives the broadcast message and rebroadcasts the message as *{RREQ, h^3(r), (A, B), (M_A,*

$M_B)\}$. Finally Node C broadcasts $\{RREQ, h^4(r), (A, B, C), (M_A, M_B, M_c)\}$ which is received by the destination node D.

Problems

- ***Fabrication Attack*** **-** Because of the one way hash function used in the route discovery process, an intermediate node cannot remove a node from the node list but can always insert new nodes e.g. in the case mentioned above C can broadcast $\{RREQ, h^5(r), (A, B, X, C), (M_A, M_B, M_X, M_c)\}$. In this case the destination node D, could not be aware of the route modification since it could verify $h^5(r)$ and sends the Route Reply. To make the attack complete, Node C will reveal two Tesla keys i.e. $K_{X_{ti}}$ and $K_{C_{ti}}$ at the time of forwarding the Route Reply RREP.
- ***ROUTE REQUEST replays*** **-** In another form of attack, assume node A receives the RREQ broadcast message from S. It acts naively first and broadcasts $\{RREQ, h^2(r), (A), (M_A)\}$, but then rebroadcasts the same packet with a new unseen id. Node B receives the RREQ, checks its local table of *(initiator, id)* values from recent REQUESTs it has received, and considering it as a new RREQ, it broadcasts the RREQ. Node A can successfully replay this packet several times till the pessimistic transit time of RREQ from S to D is elapsed. This attack can effectively become an instance of Denial of Service attack on destination node D as it will process much more RREQs.
- ***Security Association Overhead*** **-** Furthermore, for every two nodes e.g. A and B who need to communicate, Ariadne assumes two secret MAC keys K_{AB} and K_{BA} shared between them. Nevertheless, two secret shared keys appear more like an overhead for secure communication among two principals in MANETs.
- ***Clock Synchronization in MANETs*** **-** Moreover, correct working of Ariadne is strongly dependent on the loose time synchronization between the communicating nodes which is indeed very difficult to achieve in MANETs. Many approaches have been proposed so far regarding achieving time synchronization in MANETs [3] but due to unpredictability and imperfect measurability of message delays and unrealistic assumptions, clock synchronization procedures are always erroneous and are vulnerable to several attacks.
- ***TESLA's susceptibility to DoS attacks*** **–** Finally, Ariadne uses TESLA, a broadcast authentication scheme, which is based upon the principle of delayed key disclosure. Due to its use of delayed key disclosure, packets must be buffered for duration of time directly related to the maximum end-to-end network delay, and as a result TESLA becomes vulnerable to DoS attacks. Qing Li et al [12] explained the problem in detail.

C. Other Protocols

- ***Secure Routing Protocol*** – SRP [6] is a light-weight security protocol that assumes a shared symmetric key between sender and the recipient. In SRP intermediary nodes are not authenticated which makes it vulnerable to many attacks. Marshall et al [5] explained flaws in SRP, including *invisible node attack* in which the malicious node does not append its own address to the route field of the SRP header. Therefore the malicious node acts as a relay for the request and reply packets, resulting in an agreement by the source and destination on a route that is dependent on the malicious node, but does not reflect that dependency.
- ***ARAN, SADSR, BSAR and SBRP*** **–** Both ARAN [7] and SADSR [8] assume a secure uncompromisable Certification Authority (CA). Consequently, any legitimate node is supposed to acquire an offline certificate from CA. Though BSAR [10] and SBRP [9] ensure a secure binding between IP addresses and keys without assuming any trusted CA or key distribution center (KDC). However, the assumption common in these protocols that certificates are bound with IP addresses is unrealistic; roaming nodes joining MANET sub-domains will be assigned IP addresses dynamically (e.g., DHCP [11]) or even randomly. Another problem with ARAN and SADSR is that a malicious node can successfully redirect the Reply Packet (REP) from a different and longer path as opposed to the one used in Route Discovery Process. For instance, in ARAN, the source node 'S' creates a Route Discovery Packet (RDP) by signing it with its secret key. The next node A once receives the RDP broadcast, will verify the signature of 'S', sign the RDP, appends its own Certificate and rebroadcast the RDP. Once Node B receives RDP, it removes Node A's certificate and signature, appends its own certificate and rebroadcasts RDP. Likewise Node C and D will do and consequently the RDP is received by destination 'X'. At this point, Node 'X' creates a signed response message REP and unicasts it toward the node from where it received RDP *(in this case Node D).* Node D signs the received REP and attaches its own Certificate. At this instant, Node D is supposed to unicast REP to Node C, but acting maliciously, it sends the REP to E. Node E verifies the signature and certificate of D and finding it a legitimate REP, unicasts it to its known predecessor *(in this case Node F).* After REP reaches the source S via G, H, B and A, the integrity check in the source is valid since D does not modify the actual REP. Figure 1 explains the process.

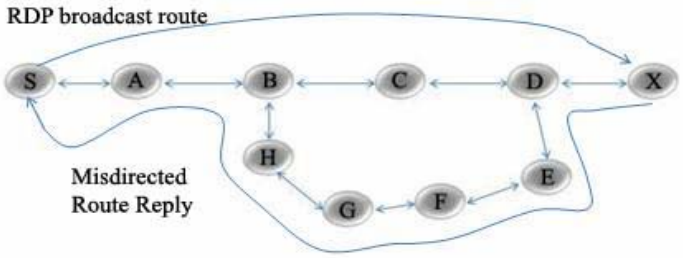

Figure 1. False Redirection of REP

III. OVERVIEW OF PARM

Our proposed scheme RSRP uses PARM [4] a lightweight, pollution attack resistant authentication scheme for broadcast

authentication. Unlike TESLA authentication protocol used by Ariadne, PARM allows the intermediate nodes to instantly authenticate the source of broadcast traffic. Hence it avoids the delay incurred in revealing the Keys in Ariadne. Moreover it is resistant to DoS attacks and does not have the buffer over flow problem prevailing in TESLA. We have customized PARM to make it suitable for MANET environment. In the subsequent discussion, we will provide an overview of the original PARM scheme.

In PARM, before communicating with receivers, the sender generates a temporal key pair, which contains a temporal secret key (TSK) chain and a temporal public key (TPK). First the sender generates 'k' n-bit random numbers (R_0, R_1,.., R_{k-1}) and denote this set of numbers as TSK_0 of the TSK chain. Subsequently, the sender uses a one-way hash function h to recursively generate the remaining TSK chain. The TSK chain has a length of L and is represented as (TSK_0, TSK_1,..,TSK_{L-1}). The temporal public key (TPK) is created by hashing every element of TSK_{L-1}. Figure 2 depicts the procedure for TSK and TPK generation. R_0 denotes the randomly generated number, and the arrows (pointing down) specify the direction of the one-way hash function h. The elements of the last row form the TPK.

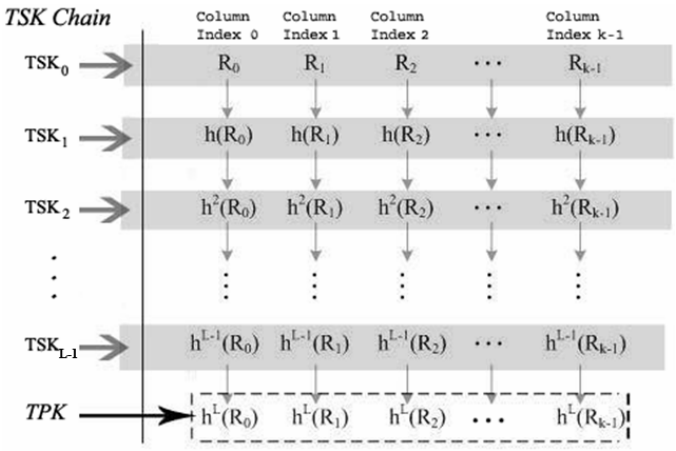

Figure 2. Temporal Key Pair Generation

After successful generation of the TSK chain and TPK, the sender provides receivers with the TPK. For a given temporal key pair, the sender maintains a usage table, such as the one in Figure 3, which tracks the number of times each column index of the TSK element array is used.

Column Index Usage	0	1	2	3	. . .	k-2	k-1
Usage Times	7	0	2	5	. . .	4	1

Figure 3. Usage Table

At this point, the sender is capable of generating the evidence which the receivers can authenticate but can't reproduce. We will not peep into how sender generates evidence and receivers subsequently validate them in PARM scheme. Since our proposed scheme, RSRP uses a customized procedure of evidence generation and validation; therefore we will discuss this matter in Section 6 i.e. Detailed Protocol Description.

IV. RSRP

We propose a Robust Source Routing Protocol (RSRP) based upon DSR that uses a lightweight, pollution attack resistant broadcast authentication protocol (PARM). RSRP defends against all the aforementioned problems and is robust in the sense that it is quicker to identify any malicious behavior, thus facilitates network nodes to drop invalid or corrupted packet earlier.

A. Basic Assumptions

We assume that all nodes communicate bi-directionally between each other and a key pre-distribution scheme distributes a set of shared secret key among every pair of nodes before the actual communication. The two nodes can negotiate a shared secret key, e.g. via the Elliptic Curve Diffie-Hellman algorithm [13] or Aldar Chan's [14] proposed distributed symmetric key management scheme for MANETs that does not require any infrastructure support. As our proposed scheme uses PARM with some customizations for mobile ad hoc environment, we inherit its assumptions e.g. all intermediate nodes maintain the index usage table.

B. Overview

Our work provides a novel approach to secure route discovery operation for MANET routing protocols. RSRP combats attacks that disrupt the route discovery process and guarantees, under the aforementioned assumptions, the attainment of correct topological information. It also abets the network nodes to dissuade malicious attempts to degrade network performance, mutilate route discovery process and flood undesirable traffic on the network. The design goals of RSRP are stated as:

- The destination node can authenticate the source node
- The destination node can authenticate every intermediate node listed in the packet header
- The source and destination node can confirm the correctness of node's sequence in the node list
- Source node can construct evidence of its identity, that all broadcast packet recipients can see but cannot reproduce.

As our protocol originates from the basic DSR functionality, it comprises of two main functions i.e. Route Discovery and Route Maintenance. In a secure MANET environment, broadcast and point-to-point traffic both are essential to be authenticated. Adding a Message Authentication Code MAC (computed with a shared secret key) to a message can assure a secure point-to-point communication. However, in the case of broadcast communication a sender node cannot compute separate MAC of every potential recipient node. Therefore, in such a case a secure broadcast authentication mechanism is required where a sender can construct evidence which the broadcast recipient nodes can authenticate. For this purpose, we have customized the PARM broadcast authentication scheme to enable broadcast communication recipients to confirm the identity of the sender.

V. DETAILED PROTOCOL DESCRIPTION

A. RSRP Route Discovery (RRD)

We now explain in detail the *RSRP* route discovery process in detail. The format of the RRD packet is as follows:

{RRD, source, destination, id, t_i, hash, node list, MAC list, E}

Once a node *S* wants to send a packet to a destination node *D* whose route information is either not present in sender's routing table or has expired; it initiates the *RRD* process by first constructing the evidence. We assume that the sender already possesses a Temporal Secret Key chain (TSK) and Temporal Public Key *(TPK)* as shown in figure 2. The sender first hashes the *TPK* with a one-way hash function *h*. The hash function is divided in *k b*-bit segments $T = \{i_0, i_1, \ldots, i_{k-1}\}$ where $2^{b-1}=k$ *(i.e. the total number of columns in TSK chain matrix)*. Now for each segment *i* comprising an integer between *0* and 2^{b-1}, that represents a column index in the Usage Table as shown in Figure 3, S determines its usage time a_i. Given the Usage Time a_i, sender determines the corresponding TSK row by selecting $TSK_{(L-1)-a_i}$. As an example, if the given index has never been used then $a_i=0$, consequently sender selects $TSK_{(L-1)}$ row. Note that $TSK_{(L-1)-a_i}$ helps to select the TSK row but we have to select one element from this complete row. Therefore sender selects the *i*-th element from this row. Thus if this was the *zero-th* segment of the set T, then sender selects the zero-th element from the TSK row i.e. in our case $h^{L-1}(R_0)$ as the first element of the evidence. Similarly, the sender computes the remaining evidence elements i.e. $E = \{e_0, e_1, .., e_{k-1}\}$ as shown in the Fig 4.

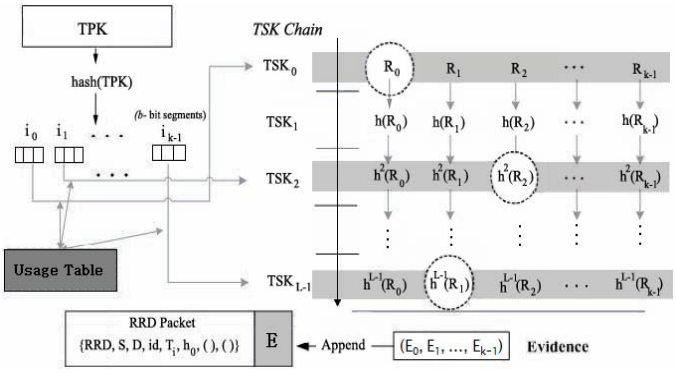

Figure 4. RSRP evidence

After constructing the evidence, sender *S* computes initializing hash h_o and broadcasts the RRD packet containing hash and evidence. Once the first hop neighbor receives the broadcast RRD packet, it can use the TPK to immediately check the validity of the attached evidence. Like the sender, the receiver must also maintain a usage table as shown in Figure for each column index of the TSK element array based on received packets. The procedure of evidence validation phase depicted in figure 5 below is just a little different from the evidence generation phase. The receiver node separates the evidence, denoted $E = \{e_0, e_1, \ldots, e_{k-1}\}$, from the RRD packet. To authenticate the evidence, the receiver hashes TPK with one-way hash function *h* used by the sender in the evidence generation phase. Next, the receiver divides the hash value

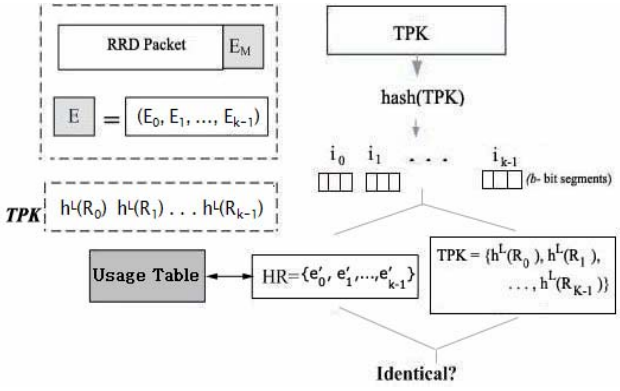

Figure 5. RSRP evidence validation phase

into *k b*-bit segments, denoting these segments as the set $(i_0, i_1, \ldots, i_{k-1})$.By interpreting each segment as an integer between 0 and 2^b-1, each segment can represent a column index of the Usage Table. Now, each index *i,* along with its usage a_i, determines the number of times to hash the corresponding element e_i of the evidence. Given the index and its usage, the receiver should perform a_{i+1} hashes on the corresponding element of the evidence. As an example, if index *i* has never been used before, the receiver only needs to perform hash on e_i once. By performing the calculated number of hashes on every element e_i of the evidence, receiver gets the hash result denoted as $HR = (e'_0, e'_1, \ldots, e'_{k-1})$. The receiver then compares the set HR with the TPK chain and considers the evidence valid if and only if the two sets HR and TPK contain identical elements. Consequently if the evidence is invalid, the receiver node will drop the RRD packet. Otherwise, it will hash the received hash, increment the usage table for the corresponding column indices, compute a Message Authentication Code (MAC) using its shared secret key with the destination node, add its id and MAC in the node list and MAC list field respectively. Finally, it broadcasts the message. Note at this point, if this intermediate node experiences a replay attempt from the last hop node *(just like the second problem mentioned in Ariadne above)*, where the intermediate node changes the id and time interval and replays the RRD packet with the same evidence; it will look into its usage table whose values are now incremented. Therefore HR will not come equal to the TPK this time, and the node will drop this packet. Thus such an attempt will not flood towards the destination node as in Ariadne. After several such RRD broadcasts the packet will finally reach the destination node. Figure 6 explains the RRD process.

As a matter of fact, it is not required for the destination to check the packet's evidence because it will first re-compute the initializing hash h_0 and compare it with the one received by the sender, if it matches then the sender is automatically authenticated. Subsequently, the destination node confirms the validity of the source route information making sure that no extra node is added, no required node is deleted and the sequence is preserved. Figure 8 illustrates the pseudo code.

RSRP Route Discovery	RSRP Route Reply
S: $h_0 = MAC_{KSD}$ (RRD, S, D, id, Ti) E = {e_0 , e_1 ,..., e_k } ▸ *S→ * {RRD, S, D, id, Ti, h_0 , ( ), ( ), E}* A: $h_1 = H[A, h_0]$ $M_A = MAC_{KAD}$ (S, D, id, h_0 , Ti)	D: $M_D = MAC_{KSD}$ (RREP, D, S, Ti, (A, B, C), (M_A , M_B , M_C)) ▸ *D → C: {RREP, D, S, Ti, (A, B, C), (M_A , M_B , M_C), M_D }*
▸ *A→ * { RRD, S, D, id, Ti, h_1 , (A), (M_A), E }* B: $h_2 = H[B, h_1]$ $M_B = MAC_{KBD}$ (S, D, id, h_1 , Ti)	▸ *C → B: {RREP, D, S, Ti, (A, B, C), (M_A , M_B , M_C), M_D}*
▸ *B→ * { RRD, S, D, id, Ti, h_2 , (A, B), (M_A , M_B), E }* C: $h_3 = H[B, h_2]$ $M_C = MAC_{KCD}$ (S, D, id, h_1 , Ti)	▸ *B → A: {RREP, D, S, Ti, (A, B, C), (M_A , M_B , M_C), M_D}*
▸ *C→ * { RRD, S, D, id, Ti, h_3 , (A, B, C), (M_A , M_B , M_C), E }* // At this point D receives the Packet	▸ *A → S: {RREP, D, S, Ti, (A, B, C), (M_A , M_B , M_C), M_D}*

Figure 6. RSRP Route Discovery and Route Reply Process

Symbol	Explanation	Symbol	Explanation
S:	Node S computes	*RRD*	RSRP Route Discovery
*S→ **	Node S broadcasts	*RREP*	RSRP Route Reply
T_i	Time Interval	$Node_i$	*I*-th node in the Node List
K_{SD}	Shared Key between Node S & D	$GetKey(Node_i)$	Get my nodes shared key with $Node_i$
id	Random packet id	a_i	Usage times of TPK column index

Figure 7. Legend

```
for( i=0 → j)   //loop begins   (See Figure. 7 for the symbols and functions)
    // in case of figure 6 it is '3' i.e. max hash value
            if(i equals 0)
                Compute h0 = MACKSD (RRD, S, D, id, Ti)
            Else
                hi = H[Nodei ,hi-1 ]
                KeyiD = GetKey(Nodei)
                MACNODEi = MACKeyiD (S, D, id, hi-1 , Ti)
                        if( MACNODEi  not equal to Given Mac)
                                check = false
                                break
    increment j  // out of the loop
    if(check equals true)
            if(hi not equal to Received Hash)
                        check = false     // Drop the RRD
    else
    Create RREP
```

Figure 8. Destination Node's Verification

B. *RSRP Route Reply (RREP)*

Once all the aforementioned checks are successful, the destination node produces the RREP. The RREP packet format is as follows

{RREP, destination, source, T_i, (node list), (MAC list), authentication MAC}

For this purpose, the destination node first calculates the authentication MAC M_D, appends it in the RREP packet and sends it to the last node in the node list. Figure 6 demonstrates the RSRP Route Reply Process. An intermediate node once receives the RREP packet checks the node list whether it is listed in it or not. If it's not listed in the node list then it discards the packet, otherwise it sends it to the preceding node in the node list. Finally the source node receives the RREP; it re-computes the authentication MAC M'_D by picking up all the fields from the RREP packet and using its shared key with the destination MAC_{KSD}.

$M'_D = MAC_{KSD}$ *(REQ, D, S, Ti, (A, B, C), (M_A , M_B , M_C))*

If the recomputed authentication MAC is equal to M_D, it means that the integrity of the packet is preserved; thus source node adds this new routing information to its routing table.

C. *Route Maintenance*

Route Maintenance in RSRP is based on DSR, thus a node forwarding a packet to the next hop along the source route returns a ROUTE ERROR (RERR) to the original sender of the packet if it is unable to deliver the packet to the next hop after limited number of retransmission attempts. In this section, we will describe the secure Route Maintenance mechanism, however we do not assume the case of an attacker not sending or forwarding the RERR message.

Once a node encounters a broken link, it creates a RERR message, appends an evidence for its identity and returns this new packet to the source of the original packet. Now all intermediate nodes see this RERR message, authenticate the sender of the error message and update their routing tables. The format of the Route Error message is shown as follows:

{RERR, packet source, attempting node, unreachable node, T_i, E}

Here *packet source* represents original sender of the packet, *attempting node* is the one that encountered problem forwarding the packet, T_i represents the time interval and *E* is the evidence of attempting node's identity. Once the RERR packet is received by the packet source, it either tries to find an alternate path or initiates a new Route Discovery.

VI. PROTOCOL ANALYSIS

Our work provides a novel approach to the secure route discovery operation for MANET routing protocols. It achieves the design goals mentioned in section 5.2 and withstands the problems mentioned in the existing protocols. We now consider scenarios where the aforementioned protocols failed to deal with and see how RSRP resists such attempts.

- ***Source Route Modification or Fabrication Attack*** - From Figure 6 we assume that in RSRP Route Reply, Node C *(just like the fabrication attack mentioned in problems in Ariadne)* broadcasts *{ RRD, S, D, id, Ti, h_4 , (A, B, X, C), (M_A , M_B , M_X ,M_C), E }*; thus adds an extra node in the node list. But to compute M_X, C needs to know the secret key K_{XD} which the node X shares with the destination. Otherwise the destination node will invalidate the MAC and discard the RRD packet.
- ***Forging Route Error Messages*** – Route Error messages cannot be forged in case of RSRP; in view of the fact that the sender is supposed to append evidence of its identity with the RERR packet.
- ***ROUTE REQUEST replays*** - Assume that node A receives the RRD packet and broadcasts it. Node B will receive the broadcast, extract the evidence of the packet and by means of its Usage Table verify the identity of S. As soon as it authenticates the evidence, it will increment the respective columns in the Usage Table. At this instant, if node A changes the id and T_i of the packet and replays the RRD packet B will extract the evidence and verify it with the Usage Table but since the Usage Table is already incremented HR will not be equal to TPK. Therefore unlike

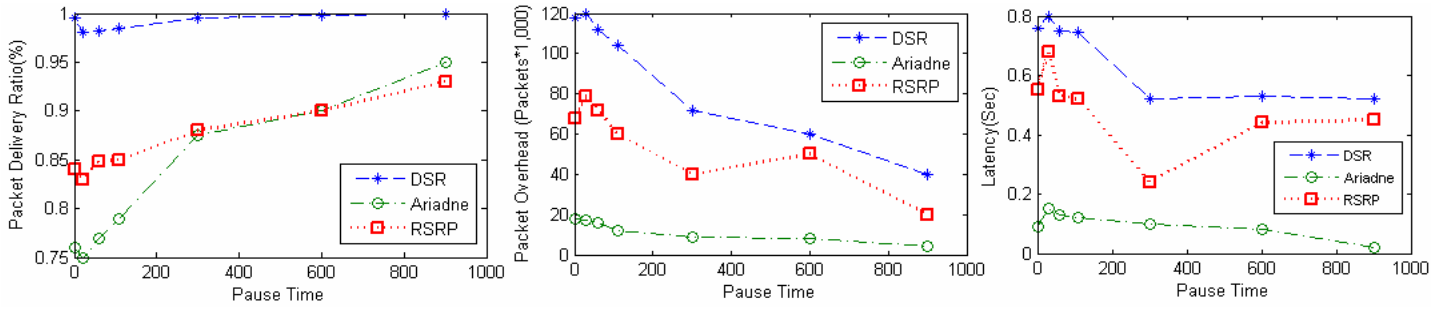

Figure 9. Packet Delivery Ratio, Routing/Packet overhead and Average Latency comparision of DSR, Ariadne and RSRP

Ariadne, it will not broadcast the replayed Route Request.

- ***Clock Synchronization & DoS attacks*** - RSRP uses a broadcast authentication mechanism that does not depend on any clock synchronization. Moreover, in view of the fact that it does not require any buffer and facilitates the broadcast recipients to instantly authenticate the sender, it is much less vulnerable to DoS attacks.
- ***Invisible Node Attack*** - Assume from Figure 1, Node C acts as an invisible node and blindly relays the broadcast packet; therefore X will receive the node listing as *(A, B, D)*. Node X will authenticate the MACs in the MAC list and will generate the RREP, but since RSRP follows strict source routing, Node D will try to send the reply to B and not to the invisible node C. Consequently D will report Route Error as it will find B unreachable.
- ***False Redirection of Route Reply*** - The problem stated in ARAN and SADSR is not possible in RSRP because RSRP ensures the integrity of nodes listed in the source route. Therefore like in Figure 1, if the node D wants to redirect the Route Reply to the false path then it has to send *{RRD, S, D, id, Ti, h_4 , (A, B, H, G, F, E, D), (M_A , M_B , M_H , M_G , M_F , M_E ,M_D), M_X, E}* instead of *{RREP, D, S, Ti, (A, B, C, D), (M_A , M_B , M_C, M_D), M_X}*.And as D cannot compute M_H , M_G , M_F , M_E therefore this attack is not possible.

Figure 9 demonstrates the comparison between RSRP, DSR and Ariadne in terms of Packet Delivery Ratio, Routing/Packet Overhead and Average Latency as a function of Pause Time where after the particular Pause time the node will change itself to any random location. Here we can see that RSRP depicts less packet overhead as compared to Ariadne, mainly because in Ariadne source has to periodically broadcast the new Tesla keys. Moreover, RSRP has less average latency because it provides instant authentication of RRD packets whereas in Ariadne, nodes wait till sender reveals the Tesla key. Hence we conclude that the resource consumption of RSRP is nevertheless modest and inevitable in order to ensure a completely secure ad hoc environment.

VII. Conclusion

Existing MANET routing protocols besides having certain advantages are subject to a variety of attacks that can disrupt and degrade the routing. We have explicated the problems in earlier routing protocols such as DSR, Ariadne, SRP, ARAN, SADSR, BSAR and SBRP. Then we introduce RSRP, a secure routing protocol based on DSR, which uses a broadcast authentication scheme that unlike Ariadne does not depends on clock synchronization and provides instant authentication. In our security analysis we have explained how certain security features of RSRP; eradicate the problems in aforementioned protocols and guarantee secure routing information acquisition.

Acknowledgment

This research was supported in part by the MIC (Ministry of Information and Communication), Korea, under the ITRC (Information Technology Research Center) support program supervised by the IITA (Institute of Information Technology Advancement) (IITA-2006-(C1090-0602-0011)).

References

[1] Yih-Chun Hu, Adrian Perrig and David B. Johnson, **"**Ariadne: a secure on-demand routing protocol for ad hoc networks". *MobiCom 2002*, Atlanta, Georgia, USA, September 2002.

[2] A. Perrig, R. Canetti, D. Song and J.D. Tygar. "Efficient and secure source authentication for multicast", in *Proceedings of the Network and Distributed Syste-m Security Symposium, NDSS'01 (February 2001)*.

[3] Kay Römer, "Time synchronization in ad hoc networks". *Proceedings of the 2nd ACM international symposium MobiHoc '01*.

[4] Ya-Jeng Lin, Shiuhpyng Shieh and Warren W. Lin. "Lightweight, pollution-attack resistant multicast authentication scheme", *ASIACCS '06*, Taipei, Taiwan, Mar 2006.

[5] J. Marshall, V. Thakur, and A. Yasinsac, "Identifying Flaws in the Secure Routing Protocol", *Proceedings of The 22nd International Performance, Computing, and Communications Conference*, April 2003.

[6] P. Papadimitratos, Z. J. Haas, and P. Samar, "The Secure Routing Protocol (SRP) for Ad Hoc Networks," *draftpapadimitratos- secure-routing-protocol-00.txt*, Dec. 2002.

[7] K.Sanzgiri, 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.

[8] S.Ghazizadeh, O.Ilghami, and E.Sirin, "Security-aware adaptive dynamic source routing protocol", *27th Annual IEEE Conference on Local Computer Networks*, 2002.

[9] Y.-C. Tseng, J.-R. Jiang, and J.-H. Lee. Secure bootstrapping and routing in an IPv6-based ad hoc network. *In ICPP Workshop on Wireless Security and Privacy*, 2003.

[10] R. Bobba, L. Eschenauer, V. Gligor, and W. Arbaugh. Bootstrapping security associations for routing in mobile ad-hoc networks, May 2002.

[11] R. Droms, "Dynamic Host Configuration Protocol", *IETF RFC-2131*, March 1997**.**

[12] Qing Li and Wade Trappe, Rutgers, "Staggered TESLA: A Multicast Authentication Scheme Resistant to DoS Attacks", *in the proceedings of IEEE GLOBECOM*, Nov 2005

[13] R. Zucceratto, and C. Adams, "Using Elliptic Curve Difffie Hellman in the SPKM GSS-API", *Internet Draft, IETF*, Aug 1999

[14] Aldar C-F. Chan, "Distributed Symmetric Key Management for Mobile Ad hoc Networks", *IEEE INFOCOM* 2004