

An Analysis of SRP for Mobile Ad Hoc Networks

John Marshall
Department of Computer Science
Florida State University
Tallahassee, FL 32306-4530

May 17, 2002

Abstract *We analyze the Secure Routing Protocol (SRP) introduced by Papadimitratos and Haas as a means for securing ad hoc networks. Flaws in the authors' conclusions are discussed pertaining to SRP's stated goals. We then introduce an attack which demonstrate SRP's vulnerabilities and propose a solution based on the watchdog scheme.*

Keywords: ad hoc networks, routing protocols, security.

1 Introduction

As its applications materialize, research concerned with *Mobile Ad Hoc NETWORKS (MANET)* has increasingly populated the literature. The emergence of mobile and wireless communications as an attractive alternative to traditional communication media has provided the impetus behind this work, in addition to military and numerous other uses [1, 2]. As we become more familiar with and acclimatized to this revolutionary communications paradigm, people at large will naturally demand more from this technology.

With this in mind, herein we analyze the *Secure Routing Protocol (SRP)* [3] as a means for securing MANET. SRP was designed to be used as an extension for existing, non-secure MANET routing protocols. Papadimitratos and Haas assert that at the completion of an SRP route discovery, the source can be **guaranteed** that the identified *route* is non-

corrupted. If SRP does not precisely meet this lofty goal then the implications for users of SRP are noteworthy. As a security add-on, SRP does not uphold its end of the bargain. The focus of our discussion is that the authors suggest numerous attack scenarios on SRP but overlook the probable one we propose.

Section 2 begins the discussion with the assumptions and necessary details of SRP. In Section 3 we discuss the weaknesses of SRP along with a critique of the designers' BAN analysis. In Section 4 we introduce our attack and propose a solution in Section 5. The paper concludes with a discussion and future work.

2 Assumptions & background

SRP was designed in the likeness of the *Dynamic Source Routing (DSR)* [1] protocol for wireless ad hoc networks introduced by Johnson and Maltz of CMU. The design was such as to minimize the assumptions about the participating nodes while guaranteeing that malicious activity during route discovery is detected. To this end, the assumptions are three-fold:

1. communication between nodes is *bi-directional*, defined in [2] to mean that communication between nodes is symmetric. That is, if A can receive from B at time t then the converse also holds (namely, B can receive from A at time t);
2. a *Security Association (SA)* exists between the source S and destination T . The

SA includes a shared key $K_{S,T}$ between the two nodes; and

3. the protocol is run in an environment with non-colluding nodes.

We claim that the strong wording (“guarantee”) is unwarranted, but we will return to this later after providing a concise description of SRP by highlighting its fundamental steps.

The following is intended only as a working knowledge of SRP; for a full description we recommend reading [3]. SRP is broken into six components: (1) route request, (2) query propagation, (3) route reply, (4) reply validation and (5) route maintenance. We will discuss each of these in turn.

2.1 Route request

For a source node S to initiate a route discovery to target node T , it places the following information in the SRP header of a query packet:

1. Q_{seq} \rightarrow monotonically increasing *Query sequence number* S maintains for each T to which it wishes to establish a path. This is used by T to recognize outdated or replayed route requests.
2. Q_{ID} \rightarrow random 32-bit number used by intermediate nodes to distinguish relayed requests.
3. *Message Authentication Code (MAC)* \rightarrow 96-bit keyed hash of the IP header, Q_{seq} , Q_{ID} and the shared key $K_{S,T}$. Any fields within either header that may change during packet propagation are excluded¹.

The *route* field is initialized with the source address S . The query packet then contains the SRP header, the underlying ad hoc routing protocol header and the IP header and it is thus transmitted.

¹This point looms large over the authors’ claim of rendering “the scheme efficient and scalable” but opens the door for our proposed attack.

2.2 Query propagation

An intermediate node I extracts the source and destination addresses from the IP header, as well as Q_{ID} from the SRP header. I uses the source and destination addresses to maintain an entry in its query table with Q_{ID} as an attribute. If an entry with a matching Q_{ID} exists, the packet is discarded; otherwise, the packet is re-broadcast with I ’s IP address appended to the *route* field.

Additionally, each intermediate node maintains a measure of the frequency of queries received from its neighbors. Incoming packets are serviced in round-robin fashion with neighbors having the lowest frequency of query transmissions receiving highest priority. It was the belief of the designers that malicious nodes would provide a much higher volume of forwarded queries. While we partially agree with this statement (in fact, this is one mechanism that might signal our proposed attack), we point out that a malicious node who drops packets would also receive the highest priority according to this scheme.

2.3 Route reply

Target node T first checks the source address from the IP header to determine if it shares an SA with the query initiator. If it does, T compares Q_{seq} from the SRP header to S_{max} , the maximum sequence number T maintains for each of its SA partners. If $Q_{seq} \leq S_{max}$, the query packet is discarded. Otherwise, T uses $K_{S,T}$ to compute the MAC with the respective header fields; the result of this computation is then compared against the MAC included in the query packet’s SRP header to establish *route* authenticity.

After these initial verifications, T is ready to compose a response. The response packet includes all the information from Section 2.1 with the inclusion of the *route* field in the SRP header’s MAC.

2.4 Reply validation

Upon receipt of a response to its route request packet to T , S verifies the source, destination, Q_{seq} and Q_{ID} fields to establish the packet’s legitimacy. Next, S compares the accumulated *route* to the reverse of the *route* field². If the two routes match, S computes the MAC from the pertinent fields and performs a comparison as T does in Section 2.3. If the two MACs are identical, S accepts the reply as non-corrupted and thus the route’s legitimacy.

2.5 Route maintenance

As the designers state quite well, “Topology changes have to be detected and the sources of the affected routes have to be notified, while avoiding false or fabricated notifications” [3]. However, it is unclear how SRP performs this functionality. More importantly, it is questionable if SRP *could* provide route maintenance without an SA between the detecting intermediate node and the source. Regardless, the authors come to the conclusion that malicious nodes can only harm routes to which they explicitly belong. In our attack (see Section 4) we show that this assertion is inaccurate.

3 Potential weaknesses of SRP

We argue that using only a MAC and an SA between two end nodes does not ensure the identified *route* between S and T actually exists. The weakness occurs for two related reasons: 1) the intermediate nodes do not have to append their address and 2) as a result, T has no way of authenticating the *route*.

We elaborate on these more when we introduce our attack in Section 4. We segue into our attack by placing it on a formal foundation. We do this by presenting the authors’ BAN analysis and critiquing their findings.

²Note that bi-directionality of node communication is necessary for use of the reverse-route to be meaningful

3.1 The authors’ BAN analysis

BAN logic [4] is a logic of belief used to evaluate security protocols. Papadimitratos and Haas provide a BAN analysis of SRP in an attempt to prove that the protocol meets its security goals. In the presentation of their analysis below, it is assumed the reader is familiar with the language of BAN logic. We provide a summary table of notation in Table 1.

Notation	Description
$P \triangleleft X$	P is told X
$P \ni X$	P possesses X
$P \sim X$	P once said X
$P \equiv X$	P believes X
$P \equiv \#X$	P believes X is fresh
$P \Rightarrow X$	P has jurisdiction over X

Table 1: BAN logic notation.

The authors break SRP into its two fundamental messages: the query request and the request response. We can represent this by:

$$\begin{aligned}
 S \rightarrow T & \quad Q_{S,T}, H(Q_{S,T}, K_{S,T}) \\
 T \rightarrow S & \quad R_{S,T}, route, H(R_{S,T}, route, K_{S,T})
 \end{aligned}$$

where $Q_{S,T}$ is the route request header information including the source, target and Q_{seq} and $R_{S,T}$ is the route reply where Q_{seq} binds the reply to $Q_{S,T}$.

The authors state the initial assumptions that we summarize but do not express formally. Both S and T possess the secret key $K_{S,T}$ and believe it is a legitimate key between them. Furthermore, S believes that the newly generated sequence number Q_{seq} is fresh and T possess the list of all sequence numbers $N_{S,T}^p$ that S has sent in previous route requests. Now, when T receives the first message from S

$$\frac{T \triangleleft (Q_{S,T}, H(Q_{S,T}, K_{S,T}))}{T \ni (Q_{S,T}, H(Q_{S,T}, K_{S,T}))}$$

and

$$\frac{T \ni (Q_{S,T}, H(Q_{S,T}, K_{S,T}))}{T \ni Q_{seq}}$$

So, T uses Q_{seq} to determine the freshness of the route request. If $Q_{seq} \notin N_{S,T}^p$ then T believes the message is fresh and checks the MAC. If all checks out then

$$T \equiv S | \sim (Q_{S,T}) \text{ and } T \equiv S | \sim (H(Q_{S,T}, K_{S,T})).$$

At this point, T prepares the response packet by including the *route* field that has been appended by the intermediate nodes. Upon receipt, the following can be stated of S

$$\frac{S \triangleleft (R_{S,T}, route, H(R_{S,T}, route, K_{S,T}))}{S \ni (R_{S,T}, route, H(Q_{S,T}, route, K_{S,T}))}$$

and

$$\frac{S \ni (R_{S,T}, route, H(R_{S,T}, route, K_{S,T}))}{S \ni Q_{seq}}.$$

So, S believes the freshness of Q_{seq} so $S | \equiv \#(Q_{seq}, route)$. As a consequence,

$$S | \equiv T | \sim (R_{S,T}), S | \equiv T | \sim (H(R_{S,T}, route, K_{S,T})). \quad (1)$$

From this we give our assessment of the derivations.

3.2 Critique of BAN analysis for SRP

From Equation 1, the authors claim that “ S believes the entire route reply originates from T and is fresh and, **trivially**, that T has constructed *route*” [3] (emphasis on “trivially” ours). While we agree with the former statement that the reply originates from T , we object to the latter claim. We give a justification for our objection below.

In their seminal paper introducing BAN logic for analysis of authentication protocols, Burrows et al. note that BAN was not designed to consider the impact of cleartext messages in protocols. Specifically, they state:

The idealized protocols of the examples given ... do not include cleartext message parts ... We have omitted cleartext communication simply because it can be forged, and so its contribution to an authentication protocol is mostly one of providing hints as

to what might be placed in encrypted messages [4].

We focus, then, on the fact that the *route* field in the SRP header is sent in the clear. While use of *route* may be legitimate when linked to an encrypted part of the message, no node can claim sole responsibility for *route*’s correctness and stand-alone use.

S uses the reverse of the *route* to identify the contents of the MAC. However, T places the *route* within the MAC without having any means to verify its integrity. The wording that T has “constructed” the *route* is grossly misleading; T has simply forwarded the accumulated *route* from a query request it has erroneously validated as non-corrupted.

We believe that the derivations provided by Papadimitratos and Haas are inadequate to reach the so-called “trivial” result. We feel it would be necessary for some node (perhaps T) to have jurisdiction (control) over *route* in order for S to believe the *route* is legitimate.

According to the *jurisdiction* rule from [4], we would require that

$$\frac{S | \equiv T | \Rightarrow route \text{ and } S | \equiv T | \equiv route}{S | \equiv route}$$

for S to truly **believe** that *route* is legitimate. However, this requirement may stretch the limits of BAN logic as there is no obvious mechanism to provide jurisdiction over such a global field.

As such, the authors do not make any claims that T believes it **controls** *route*, let alone that S believes that T believes *route*. So, then, what basis does S have to believe that T has jurisdiction over *route*? We argue that no basis for such a claim exists and our attack in Section 4 exploits this perceived weakness.

4 Proposed attack on SRP

The attack we propose falls within the assumptions of Section 2. More specifically, we show that an attack is possible even when the intermediate nodes are non-colluding. Our attack

relies on a highly active malicious intermediate node.

The premise of our attack is that a malicious intermediate node may not append its address to the *route* field of the SRP header. Recall, the appended addresses are used by the target node T to authenticate a path between S and T . T correctly authenticates the query packet and sends a response back along the reverse path. The same malicious node then recognizes the packet as the response to the query to which it did **not** append its address. Again, M simply forwards the packet without appending its address. Upon receipt of the response, S calculates the MAC with the header fields including the reverse of the appended *route*. S compares this with the attached MAC and finds a match. As a result, S has unwittingly verified a path which does not exist lest M continues to forward packets.

4.1 An example

For this example, we will use the ad hoc network represented by the graph of Figure 1. Links between nodes represent the fact that the two nodes are within transmission range (i.e. assuming bi-directionality, both nodes can send and receive packets between them).

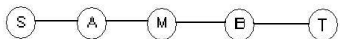


Figure 1: Ad hoc network topology for the example attack.

To clarify the notation used below, we will use $A \rightarrow B$ not to specifically mean A “sends to” B ; rather, this notation implies that A broadcasts a message which B receives. The triples given below will follow the following format: $\langle \text{sender} \rangle \rightarrow \langle \text{receiver} \rangle : \langle \text{route} \rangle$. The number before each triple naturally represents the sequence of events. First, we’ll show the events leading to an attack on the query request.

1. S creates query : calculates MAC

2. $S \rightarrow A : S$

3. $A \rightarrow M : SA$

4. $M \rightarrow B : SA$, M does not append address

5. $B \rightarrow T : SAB$

6. T verifies : route = $SABT$, T verifies MAC

T will accept the query packet as non-corrupted. Then T sends the response packet.

1. T creates response : appends route = $SABT$ and MAC

2. $T \rightarrow B : T$

3. $B \rightarrow M : TB$

4. $M \rightarrow A : TB$, again M does not append address

5. $A \rightarrow S : TBA$

6. S verifies : $TBAS = \text{reverse}(SABT)$, S verifies MAC

The final check by S passes, so S accepts the response and thus the *route* $SABT$ as legitimate. This completes the attack.

4.2 Implications

The result is that S erroneously believes a route exists between it and T which is not dependent upon M as an intermediate node. Yet, the route undoubtedly relies on M to forward packets so that they finally reach T . To illustrate the possible implication of this false belief consider what happens when M leaves the ad hoc network. Any route maintenance technique will be unable to notify S that the route is no longer intact because it is believed the route does not rely on M ’s presence.

This is but one example. What is important to note, however, is that SRP does not meet its goal of guaranteeing the identified route is non-corrupted. We now introduce a proposed solution to the attack we have presented.

5 Proposed solution

In [2], the *watchdog* tool is introduced as a means to detect and mitigate node misbehavior. Here, we propose an adaptation to *watchdog* we call *bloodhound* which detects our attack. The solution is to add *bloodhound* to SRP to make it secure against the proposed attack. First, we shall describe *watchdog* and then our adaptation to it.

5.1 Watchdog

As in Figure 1, suppose a path exists between our source and destination nodes S and T , with intermediate nodes A , M and B . Because A cannot reach B directly, it routes packets through M who is within transmission range of both A and B . Not only does B receive this packet but, due to the bi-directionality assumption, A overhears M forward. In this way, A determines if its packet has been forwarded.

Now, the authors make use of *watchdog* to determine if A 's identical packet has been forwarded by M . This is done by maintaining a buffer in A of recently sent packets. If a match is found, then M has properly handled the forward and the buffer entry is erased. On the other hand, if after a specified *timeout* period has elapsed A does not overhear its packet forwarded, then A increments a failure tally kept for M . Once the tally exceeds a threshold, M 's behavior is flagged as malicious.

5.2 Bloodhound - an adaptation to watchdog for use with SRP

The adaptation to *watchdog* we propose is minor, in that the overriding principle is maintained while the interpretation of, and response to, a match differs. Because *bloodhound* is used in conjunction with SRP we know how a "loyal" intermediate node successfully forwards a query packet (see Section 2.2). To recapitulate, an intermediate node need only append its address to the *route* field then forward the packet. As a result, A should **never** receive an identical packet to one it previously

sent.

To illustrate, let's revisit our example from Section 4. Suppose A forwards a query packet with *route* = SA . Any intermediate node I_i within A 's transmission range will receive the packet, append its address (I_i) and forward the packet. A overhears this forward and notes that *route* = SAI_i is **not** identical to the one it sent, signalling this is a correct run of the protocol forward.

Suppose, however, that the intermediate node is the malicious node ($I_i = M$) from our attack in Section 4. In this case, M will not append its address to the packet. So, M will forward exactly what it received from A . Since M cannot do a partial broadcast, A will overhear the packet and notice its identical header content.

Now, if A does encounter an entry match then it knows malicious activity is being perpetrated by one of its neighbors, although it knows not which one. A can only propagate a message back to S (if S and A have an SA and an established path) to inform S that any path between S and T upon which A lies is suspect.

6 Conclusions & future work

We introduced the SRP algorithm for routing in ad hoc networks and exposed a vulnerability by employing BAN logic. This weakness was realized with our attack from Section 4. We proposed a solution, *bloodhound*, which we feel helps mitigate the attack.

With this said, one question still looms large: "Is there a way to secure SRP without having to resort to a PKI scheme?" As the authors quite rightly state, such an approach demands much of intermediate nodes in terms of computation, group management and key management. However, the trade-offs between maintaining state in the intermediate nodes and implementing a secure PKI scheme to our knowledge have not been fully considered.

7 Acknowledgements

While this section is perhaps least significant to readers, this author would be remiss not to extend his sincerest gratitude to those without whose help this paper would not have developed. A special thanks to Stephen Carter who introduced SRP to our research group. A further thanks to the members of the research group for their insights on the protocol and the attack. Finally, I am indebted to Alec Yasinsac for his continued guidance and also his suggested revisions to this document.

References

- [1] David B. Johnson and David A. Maltz. Dynamic source routing in ad hoc wireless networks. In Tomasz Imielinski and Hank Korth, editors, *Mobile Computing*. Kluwer Academic Publishers, 1996.
- [2] Sergio Marti, T. J. Giuli, Kevin Lai, and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. *Mobile Computing and Networking*, pages 255–65, 2000.
- [3] P. Papadimitratos and Z. J. Haas. Secure routing for mobile ad hoc networks. *SCS Communication Networks and Distributed Systems (CNDS 2002)*, January 27-31 2002.
- [4] M. Burrows, M. Abadi, and R. M. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, Feb 1990.