# Routing Protocol using One-Way Hash Functions for Mobile Ad Hoc Networks

**YoungHo Park**
School of Electronics and Electrical Engineering, Sangju National University,
386, Gajang-dong, Sangju-si, Kyungsangpook-do, 742-711, Korea
Tel: +82-54-530-5323, Fax: +82-54-530-5323, E-mail: yhpark@sangju.ac.kr

**Abstract:** *An ad hoc network is a collection of mobile nodes without any infrastructure. However, ad hoc networks are vulnerable to attacks such as routing disruption and resource consumption; thus, routing protocol security is needed. This paper proposes a secure and efficient routing protocol for mobile ad hoc networks, where only one-way hash functions are used to authenticate nodes in the ROUTE REQUEST, while additional public-key cryptography is used to guard against active attackers disguising a node in the ROUTE REPLY.*

**Keywords:** *Routing protocol; Security; Ad hoc network*

## 1. Introduction

An ad hoc network is often defined as a non-infrastructure network, meaning a network without the usual routing infrastructure such as fixed routers and routing backbones. The wireless medium as well as the non-infrastructure nature of ad hoc networks makes them increasingly vulnerable to a number of attacks. Unlike wired networks where the attacker needs to gain access to the physical medium to launch any kind of attack, in the wireless case, an intruder can easily eavesdrop on the ongoing traffic. Since there is no centralized infrastructure, it is very difficult to have a key distribution center or a trusted certification authority to provide cryptographic keys and digital certificates to help nodes authenticate themselves.[1-3]

Attacks on ad hoc routing protocols generally fall into routing-disruption attacks and resource consumption attacks. In the former case, the attacker attempts to cause legitimate data packets to be routed in dysfunctional ways. An example of a routing-disruption attack is when an attacker sends forged routing packets to create a routing loop, causing packets to traverse nodes in a cycle without reaching their destinations, thereby consuming energy and available bandwidth. In the latter case, the attacker injects packets into the network in an attempt to consume valuable network resources, such as bandwidth, or to consume node resources such as memory or computation power.[2,4]

Many routing protocols for ad hoc networks have been proposed. The Ariadne protocol[5] based on DSR(dynamic source routing)[6] is a secure on-demand routing protocol that can withstand node compromise and relies only on highly efficient symmetric cryptography. This protocol uses a message authentication code(MAC) to verify the authenticity of the ROUTE REQUEST, and also uses one-way hash functions to verify that no hop was omitted. In the ARAN (authenticated routing for ad hoc networks) protocol[7], which is based on the AODV (ad hoc on-demand distance vector) protocol, each node has a certificate signed by a trusted authority, which associates its IP address with a public key. This protocol is an on-demand protocol, broken up into route discovery and maintenance. The SAODV(secure AODV) protocol[1] uses signatures to authenticate most fields of route request and route reply packets, and also uses hash chains to authenticate the hop count. The SAODV designs signature extensions to the AODV, where network nodes authenticate the AODV routing packets with a SAODV signature extension, which prevents certain impersonation attacks.

The Ariadne protocol does not guard against passive attackers eavesdropping on the network traffic, and does not prevent an attacker from inserting data packets. In addition, the Ariadne protocol is vulnerable to an active 1-1 attacker that lies along the discovered route[3]. Meanwhile, the ARAN protocol is computationally expensive in nodes, because this protocol uses public-key cryptography for authentication and is particularly vulnerable to DoS attacks based on flooding the network with bogus control packets for which signature verifications are required. As long as a node cannot verify signatures at line speed, an attacker can force that node to discard some fraction of the control packets it receives. In the ARAN and SAODV protocols, there are high loads to be managed by the route table in each node, because route request and route reply packets do not include route information.

This paper proposes a secure and efficient routing protocol for mobile ad hoc networks, where only one-way hash functions are used to authenticate nodes in the ROUTE REQUEST, while additional public-key cryptography is used to guard against active attackers disguising a node in the ROUTE REPLY. Our protocol is more secure than the Ariadne protocol and has higher network performance than the ARAN and SAODV protocols.

## 2. Ariadne Protocol

Ariadne is a secure on-demand routing protocol that withstands node compromise and relies only on highly efficient symmetric cryptography. The Ariadne protocol designs in two stages: this protocol first presents a mechanism that lets the target verify the authenticity of the ROUTE REQUEST and then presents an efficient per-hop hashing technique to verify that no node is missing from the node list in the REQUEST. In the following discussion, it is assumed that the initiator S performs a route discovery for target D and that they share the secret keys KSD and KDS, respectively, for message authentication in each direction.

To convince the target of the legitimacy of each field in a ROUTE REQUEST, the initiator simply includes a message authentication code (MAC) computed with key KSD over unique data (for example, a timestamp). The target can easily verify the route request's authenticity and freshness using the shared key KSD. In a route discovery, the initiator wants to authenticate each individual node in the node list of the ROUTE REPLY. A secondary requirement is that the target can authenticate each node in the node list of the ROUTE REQUEST so that it will return a ROUTE REPLY only along paths that contain legitimate nodes. Each hop authenticates the new information in the REQUEST using its current Tesla key. The target buffers the REPLY until intermediate nodes can release the corresponding Tesla keys. The Tesla security condition is verified at the target, and the target includes a MAC in the

REPLY to certify that the security condition was met. Figure 1 shows the Ariadne routing protocol in mobile ad hoc networks.

S: h0= MAC$_{KSD}$ (REQUEST, S, D, id, ti)
S → broadcast:< REQUEST, S, D, id, ti, h0, ( ), ( ) >
A: h1= H[A, h0]
MA = MAC$_{KAti}$ (REQUEST, S, D, id, ti, h1 , (A), ( ))
A → broadcast: < REQUEST, S, D, id, ti, h1, (A), MA) >
B: h2= H[B, h1]
MB = MAC$_{KBti}$(REQUEST, S, D, id, ti, h2 , (A, B), (MA))
B → broadcast: < REQUEST, S, D, id, ti, h2, (A, B), (MA, MB)
>
C: h3= H[C, h2]
MC = MAC$_{KCti}$((REQUEST, S, D, id, ti, h3 , (A, B, C),
(MA, MB))
C → broadcast: < REQUEST, S, D, id, ti, h3,
(A, B, C), (MA, MB, MC) >

D: MD = MAC$_{KDS}$ (REPLY, D, S, ti, (A, B,C), (MA, MB, MC))
D → C: < REPLY, D, S, ti, (A, B,C), (MA, MB, MC), MD, ( ) >
C → B: < REPLY, D, S, ti, (A, B,C), (MA, MB, MC), MD, (K$_{Cti}$)
>
B → A: < REPLY, D, S, ti, (A, B,C), (MA, MB, MC), MD, (K$_{Cti}$,
K$_{Bti}$) >
A → S: < REPLY, D, S, ti, (A, B,C), (MA, MB, MC), MD, (K$_{Cti}$,
K$_{Bti}$, K$_{Ati}$) >

**Figure 1. The Ariadne routing protocol**

## 3. ARAN Protocol

ARAN is an on-demand protocol, broken up into route discovery. Figure 2 shows an example of routing in ARAN. To initiate a route discovery, the initiator (in thisexample, S) broadcasts a signed ROUTE REQUEST packet that includes the target (D in the example), its certificate (certS), a nonce N, and a timestamp t. The nonce and timestamp together ensure freshness when used in a network with a limited clock skew. Each node that forwards this REQUEST checks the signature or signatures.

S → broadcast: < ( ROUTE REQUEST, D, certS, N, t )K$_S$ – >
A → broadcast: < (( ROUTE REQUEST, D, certS, N, t )K$_S$ – )K$_A$ –,
certA >
B → broadcast: < (( ROUTE REQUEST, D, certS, N, t )K$_S$ – )K$_B$ – ,
certB >
C → broadcast: < (( ROUTE REQUEST, D, certS, N, t )K$_S$ – )K$_C$ – ,
certC >

D → C: < (( ROUTE REPLY, S, certD, N, t )K$_D$ – >
C → B: < (( ROUTE REPLY, S, certD, N, t )K$_D$ – )K$_C$ –, certC >
B → A: < (( ROUTE REPLY, S, certD, N, t )K$_D$ – )K$_B$ –, certB >
A → S: < (( ROUTE REPLY, S, certD, N, t )K$_D$ – )K$_A$ –, certA >

**Figure 2. The ARAN protocol**

## 4. SAODV Protocol

The idea behind SAODV is to use a signature to authenticate most fields of a ROUTE REQUEST and ROUTE REPLY and to use hash chains to authenticate the hop count. SAODV designs signature extensions to AODV. Figure 3 shows an example of route discovery in SAODV. When forwarding a ROUTE REQUEST in SAODV, a node first authenticates the ROUTE REQUEST to ensure that each field is valid. It then performs duplicate suppression to ensure that it forwards only a single ROUTE REQUEST for each route discovery. The node then increments the hop-count field in the ROUTE REQUEST header, hashes the hop count authenticator, and rebroadcasts the ROUTE REQUEST, together with its ROUTE REQUEST - SSE extension. When the ROUTE REQUEST reaches the target, the target checks the authentication in the ROUTE REQUEST- SSE. If the ROUTE REQUEST is valid, the target returns a ROUTE REPLY as in AODV.

S → broadcast: < ( ROUTE REQUEST, id, S, seqS, D, oldseqD, h0, N ) $K_{S-}$, 0, $h_N$ >
A → broadcast: < ( ROUTE REQUEST, id, S, seqS, D, oldseqD, h0, N ) $K_{S-}$, 1, $h_{N-1}$ >
B → broadcast: < ( ROUTE REQUEST, id, S, seqS, D, oldseqD, h0, N ) $K_{S-}$, 2, $h_{N-2}$ >
C → broadcast: < ( ROUTE REQUEST, id, S, seqS, D, oldseqD, h0, N ) $K_{S-}$, 3, $h_{N-3}$ >

D → C: < ( ROUTE REPLY, D, seqD , S, lifetime, h‘0 , N ) $K_{D-}$, 0, $h'_N$ >
C → B: < ( ROUTE REPLY, D, seqD , S, lifetime, h‘0 , N ) $K_{D-}$, 1, $h'_{N-1}$ >
B → A: < ( ROUTE REPLY, D, seqD , S, lifetime, h‘0 , N ) $K_{D-}$, 2,$h'_{N-2}$ >
A → S: < ( ROUTE REPLY, D, seqD , S, lifetime, h‘0 , N ) $K_{D-}$, 3, $h'_{N-3}$ >

**Figure 3. The SAODV protocol**

## 5. The Proposed Routing Protocol

In the following, it is assumed that the initiator S performs a route discovery for target D and they share the secret keys KSD and KDS, respectively, for message authentication in each direction. To convince the target of the legitimacy of each field in the ROUTE REQUEST and the ROUTE REPLY, we simply include a message authentication code computed with key KSD. In the route discovery, the initiator wants to authenticate each individual node in the node list of the ROUTE REPLY. The target can authenticate each node in the node list of the ROUTE REQUEST so that it will return a ROUTE REPLY only along paths that contain legitimate nodes. One-way hash functions are used to verify that no node was omitted. To change or remove a previous node, an attacker must either hear a REQUEST without that node listed or must be able to invert the one-way hash function. Public-key cryptography is also used in the ROUTE REPLY to guard against active attackers disguising a node on the network.

$$\begin{array}{l}
S : h_0 = MAC_{K_{SD}}(REQUEST\ , S, D, id, ti) \\
S \rightarrow broadcast\ : \langle REQUEST\ , S, D, id, ti, h_0, (\ ) \rangle \\
A : h_1 = H[A, h_0] \\
A \rightarrow broadcast\ : \langle REQUEST\ , S, D, id, ti, h_1, (A) \rangle \\
B : h_2 = H[B, h_1] \\
B \rightarrow broadcast\ : \langle REQUEST\ , S, D, id, ti, h_2, (A, B) \rangle \\
C \rightarrow h_3 = H[C, h_2] \\
C \rightarrow broadcast\ : \langle REQUEST\ , S, D, id, ti, h_3, (A, B, C) \rangle
\end{array}$$

$$\begin{array}{l}
D : h_0' = MAC_{K_{DS}}(REPLY\ , D, S, ti) \\
D \rightarrow C : \langle (REPLY\ , D, S, ti)_{K_D -}, h_0', (A, B, C) \rangle \\
C : h_1' = H[C, h_0'] \\
C \rightarrow B : \left\langle ((REPLY\ , D, S, ti)_{K_D -})_{K_C -}, h_1', (A, B, C) \right\rangle \\
B : h_2' = H[B, h_1'] \\
B \rightarrow A : \left\langle (((REPLY\ , D, S, ti)_{K_D -})_{K_C -})_{K_B -}, h_2', (A, B, C) \right\rangle \\
A : h_3' = H[A, h_2'] \\
A \rightarrow S : \left\langle \left( (((REPLY\ , D, S, ti)_{K_D -})_{K_C -})_{K_B -} \right)_{K_A -}, h_3', (A, B, C) \right\rangle
\end{array}$$

**Figure 4. The proposed routing protocol**

Figure 4 shows the secure route discovery protocol for ad hoc networks. The route request packet consists of seven fields (ROUTE REQUEST, initiator, target, id, time interval, hash chain, node list). The initiator and target are set to the addresses of the source and destination respectively. The time interval is set to the Tesla time interval at the pessimistic arrival time of the REQUEST at the target, with maximum possible clock offset/skew and the maximum transmission delay. When an intermediate node A receives a ROUTE REQUEST, the node checks its local table initiator. If it finds an entry for the same route discovery, it discards the REQUEST, otherwise node A verifies the time interval of the REQUEST. If the time interval is too great, the REQUEST is discarded, otherwise the node appends its address A to the node list in the REQUEST, and the hash chain field is replaced by H(A; old hash chain).

When the target node receives the REQUEST, it checks whether the time interval and the hash chain are correct. If these two values are satisfied, the REQUEST is deemed valid and the target constructs a REPLY to the initiator. The route reply packet consists of three fields (encryption field, hash chain, node list). The REPLY, target, initiator and time interval are encrypted using the secret keys of each node. This encryption field is then verified by the public keys of each node in the source route, thereby guarding against active attackers disguising a node on the network. The MAC is calculated over four fields (REPLY, target, initiator and time interval) using the key KDS. The REPLY is then returned to the initiator along the source route, which is the reverse of the sequence of nodes in the node list in the REQUEST. When an intermediate node C receives a ROUTE REPLY, the node encrypts the encryption field using its secret key and replaces the hash chain field. When the initiator receives the REPLY, it decrypts the encryption field using the public keys of the node list and checks whether the time interval and the hash chain are correct. Only if all these are valid, will the initiator accept the ROUTE REPLY.

## 6. Conclusions

This paper proposed a secure routing protocol for mobile ad hoc networks. In the round trip of the proposed protocol, an intermediate node has computational loads for a one-way hash function used twice and a time public-key cryptography. Thus, it is computationally less expensive than the ARAN and SAODV protocols. The ARAN and SAODV protocols both have high loads to be managed by the route table in each node, because the route request and route reply packets do not include route information. In contrast, the proposed protocol includes a node list in the ROUTE REQUEST and ROUTE REPLY, resulting in a small load for each node.

## References:

[1] M. Guerrero Zapata and N. Asokan, "Securing Ad Hoc Routing Protocols," Proc. ACM Workshop on Wireless Security, Atlanta, Georgia, USA, pp.1-10, September 2002.

[2] YoungHo Park et al., "Secure route discovery protocol for ad hoc networks," IEICE Trans. Fundamentals, Vol.E90-A, No.2, February 2007

[3] S. Gupte and M. Singhal "Secure Routing in Mobile Wireless Ad Hoc Networks," Ad Hoc Networks, ELSEVIER, pp.151-174, 2003.

[4] Y. C. Hu and A. Perrig, "A survey of secure wireless ad hoc routing," IEEE Security & Privacy Magazine, Vol.2, No.3, pp.28-39, May-June 2004.

[5] Y. C. Hu, A. Perrig, and D.B. Johnson, "Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks," Proc. 8th Ann. Int'l Conf. Mobile Computing and Networking, Atlanta, Georgia, USA, pp.12-23, September 2002.

[6] D. B. Johnson, "Routing in Ad Hoc Networks of Mobile Hosts," Proc. IEEE Workshop on Mobile Computing Systems and Applications, Santa Cruz, CA, USA, pp.158-163, December 1994.

[7] K. Sanzgiri et al., "A Secure Routing Protocol for Ad Hoc Networks," Proc. 10th IEEE Int'l Conf. Network Protocols, San Antonio, TX, USA, pp.78-87, January 2002.