Safeguarding Mutable Fields in AODV Route Discovery Process

Kulasekaran A. Sivakumar, Mahalingam Ramkumar
Department of Computer Science and Engineering
Mississippi State University, Mississippi State, MS 39762.
⟨sa151,ramkumar⟩@cse.msstate.edu

Abstract— **Assuring cryptographic integrity of mutable fields in any on-demand ad hoc routing protocol is more challenging than that of non mutable fields. We propose an efficient authentication strategy for this purpose, which leverages a recently proposed broadcast encryption (BE) scheme. We investigate some shortcomings of SAODV, a popular secure extension of the ad hoc on-demand distance vector (AODV) protocol and suggest some modifications to the protocol to overcome the shortcomings. The modifications include proactive maintenance of a secure reliable delivery neighborhood (RDN) by each node and the use of the BE based authentication strategy for mutable fields.**

I. Introduction

The challenges associated with efficient protocols for cooperative routing in mobile ad hoc networks (MANETs) [1] have received substantial attention in the literature. In their original incarnations, most ad hoc routing protocols did not consider security as an issue to be addressed. All participants are implicitly trusted to perform their assigned tasks faithfully. Secure routing protocols [2] - [8] try to account for the possibility of nodes which may not follow the protocol and / or send deliberately misleading routing information. The problem of secure routing has attracted much attention during recent years. Most secure routing protocols are extensions of popular ad hoc routing protocols with some additional features to support cryptographic authentication of routing information. In this paper, we restrict ourselves to securing the route discovery process in the ad hoc on-demand distance vector (AODV) protocol.

The route discovery process in AODV involves flooding of a route request (RREQ) packet by a source, addressed to a destination, which are in turn broadcast (after some modifications) by intermediate nodes. Thus such RREQ packets contain mutable information which changes at every hop, and some non mutable information supplied by the source (and carried forward all the way to the destination). In AODV, the mutable information is a hop count, which is incremented by 1 at every node that forwards the RREQ. Authentication of mutable fields is more difficult than authenticating non mutable fields as every node performing alterations has to append some authentication, which may have to be carried over till the destination of the packet.

Notwithstanding the fact that carrying over authentication can be expensive, it still does not prevent many simple attacks involving shortening of the path by malicious nodes, or *node deletion* attacks [6]. Furthermore, in order to verify the authentication appended by every node, the destination also has to know the identity of every node in the path. The need for the destination to know the IDs of every node in the path obviously goes against the basic philosophy of distance vector protocols.

It is well known [8] that no secure routing protocol can guarantee integrity of the route discovery process in the face of colluding nodes. Thus most secure route discovery processes have to limit themselves to providing assurances against attacks by (perhaps multiple) *non-colluding* nodes. In such a scenario, carrying over authentication appended by every node to two hops is sufficient, as long as node *deletion* attacks can be prevented by some means.

There are two basic approaches to mitigate node deletion attacks: 1) the use of one-way hash chains, and 2) the use of two-hop group secrets. For example, the former approach is used in the secure AODV (SAODV) protocol in [2] and in secure dynamic source routing (DSR) protocols like Ariadne [6]. The latter approach is employed in [4]. The disadvantage of the former approach is that it does not prevent attackers from *increasing* the hop count (or node insertions in source routing based protocols). While the second approach can detect attempts to decrease or increase hop counts, maintaining a group secret with all two-hop neighbors can involve substantial overheads.

A. Our Contributions

This contributions of this paper are two-fold. The first is an efficient strategy for facilitating authentication of mutable fields to facilitate detection of *all* possible malicious modifications that can be performed by non colluding attackers. The efficiency of the proposed approach comes from the fact that nodes only have to maintain a consistent *one-hop* topology information, to permit two hop authentication. This efficient authentication strategy is made possible by the use of *broadcast encryption* (BE), a security primitive that has thus far received attention mostly in the context of digital rights management and multicast communication scenarios. More specifically the authentication employs a multi-source BE (MSBE) scheme proposed recently [14], [15].

The second contribution of this paper is a secure AODV protocol SAODV-2 very much similar to the SAODV protocol proposed by Zapata et al [2]. We highlight some security pitfalls of the SAODV protocol in [2] and propose modifica-

1-4244-1251-X/07/$25.00 2007 IEEE.

TABLE I

NOTATIONS USED

$A, B, \ldots,$	(uppercase alphabets) node IDs
$\|\|$	concatenation of fields
$K(M)$	symmetric encryption of a message M using a key K
K_{SD}	shared symmetric key between S and D
$h()$	cryptographic hash function
$h^i()$	repeated application of the hash function $h()$, i times
$h(M, K)$	Hashed message authentication code (HMAC) for a message M using the secret K
$\mathbb{N}_A$	set of one hop neighbors of A in the reliable delivery neighborhood (RDN) of A
K_A	secret provided by node A to all its one-hop neighbors (members of the set $\mathbb{N}_A$)
T_A	secret chosen by A that is explicitly *protected* from all one-hop neighbors (all nodes in A's RDN)
$rreq$	non mutable fields of an RREQ message
$rrep$	non mutable fields of an RREP message

tions to prevent such attacks. The modified SAODV-2 protocol presented in this paper employs MSBE for authentication of mutable fields.

Section II provides a brief overview of AODV and issues in securing non mutable fields of RREQ packet in AODV. Section II also describes some of the pitfalls of approaches in current literature, with more focus on SAODV [2]. Section III provides an overview of a recently proposed [14], [15] multi-source broadcast encryption (MSBE) scheme and discusses some of its unique features that make it very well suited for two-hop authentication in AODV. Section IV presents the SAODV-2 protocol. Conclusions are offered in Section V. A list of commonly used notations in this paper are summarized in Table 1.

II. SECURING AODV

AODV is an on-demand extension of the dynamic sequenced distance vector (DSDV) [9] protocol. When a node finds that it does not have a route to some destination, it originates the route discovery process by broadcasting a route request (RREQ) packet. This RREQ packet includes source ID, destination ID, a sequence number of the source, a last known sequence number of the destination and the maximum number of hops the RREQ can be forwarded. Any intermediate node receiving this request checks whether it has already seen the request. If so, it drops the packet. If the packet has not been seen before, it increments the hop count by one and rebroadcasts the packet.

If an intermediate node has the path to the destination with a sequence number equal to or greater than the last known sequence number indicated by the RREQ source it generates a route reply (RREP) packet. Otherwise, it just stores the information about the (previous hop) neighbor from which it received the packet. This information will be used during the route reply process. A destination node receiving the RREQ generates a RREP packet by copying all the information from RREQ packet and updating its sequence number in the RREP packet. This RREP packet is unicast back to the source node.

The first step towards securing route discovery process is the addition of the ability to provide cryptographic authentication of mutable and non-mutable fields. Typically RREQ packets from the source, or more specifically the non mutable portions of the RREQ packet may be authenticated using digital signatures (perhaps with an appended certificate if off-line distribution of certificates is not feasible). However mandating even intermediate nodes to append digital signatures may substantially increase the overheads required.

A. SAODV

Zapata et al [2] propose secure AODV (SAODV) protocol that employs a per-hop hashing technique to protect mutable fields and a digital signature of the RREQ source for protecting non mutable fields. The non mutable fields in the RREQ includes a commitment to a random value X chosen by the RREQ source in the form of $h^{h_c}(X)$, where h_c is the maximum number of hops the RREQ can be relayed. The value h_c is also indicated in the non mutable part of the RREQ. The source releases X along with the RREQ. The nodes one hop away from the source are expected to hash the value X once and forward the value $h^1(X) = h(X)$ along with the RREQ, and increment the mutable hop count value to 1. A node two hops away similarly replaces $h^1(X)$ with $h^2(X)$ and sets the hop count to 2. Thus a node i hops away will receive an RREQ with the value $h^{i-1}(X)$, and is expected to forward $h^i(X)$ indicating a hop-count of i.

1) Attacks on SAODV:

a) Modifying Hop Count:: A node i hops away from the source of the RREQ which receives a per-hop hash value $h(i - 1(X)$ could

1) forward the RREQ without hashing the per-hop hash value once in order to reduce the total hop count by one, or
2) forward the RREQ and increase the hop count by any number j, by indicating $i+j$ instead of i in the mutable hop-count field and appending a hash value $h^{i+j}(X)$ instead of $h^i(X)$.

b) External Attackers: In SAODV intermediate nodes that forward the RREQ do not append any kind of authentication to prove that they are indeed valid nodes eligible to take part in the network. Thus any external attacker can take part in the RREQ relaying process. While external attackers can be kept out by using a network-wide shared secret [6], any (malicious) internal node[1] can forward an RREQ (either with much longer hop count or without incrementing the hop count) and freely impersonate any other node in the network for this purpose. Such RREQs can preempt good RREQs over other paths from reaching the destination as every node forwards only one RREQ [16].

c) One-Way Links: SAODV implicitly assumes that all links are bidirectional. Often the use of MACA protocols (like 802.11) where a sender and receiver exchange RTS / CTS packets to ensure bidirectional links is offered as the justification for this assumption. However RTS / CTS

[1] Or any attacker who has access to the network-wide group secret (which arguably is very difficult to protect).

handshakes are *only possible for unicast messages* like RREP. Packets meant for multiple nodes (for example RREQ packets which are flooded and thus cannot indicate a specific receiver) do not support such handshakes. Thus RREQ packets *can* reach nodes which do not have a reverse link. Unless specific measures are taken to ensure that RREQ packets relayed by a node B cannot be forwarded by a node C which is not in the reliable delivery neighborhood[2] (RDN) of B, the RREP will fail if the destination happens to choose the RREQ through this path.

Also note that *even* if RREQ packets are unicast individually to every neighbor by a node B, it still does not prevent a node C within the transmission range of B (but cannot be heard by B) in gaining access to the RREQ packet, and more importantly the per-hop hash value in the RREQ packet [17]. Thus ensuring one-way links cannot be used for forwarding RREQs requires a more proactive approach.

B. Other Secure AODV Protocols

Pirzada et al [3] proposed a routing protocol that requires that all communications between one-hop neighbors be encrypted by using a group secret. A node A provides a secret K_A to all its neighbors. While such an approach can keep external attackers a bay, the protocol is susceptible to attacks by malicious internal nodes which can increase or decrease the hop count.

Du et al [4] employ one-hop and two-hop group secrets to facilitate two-hop authentication. In their approach nodes proactively determine the two-hop topology and securely deliver a two-hop group secret to every two-hop neighbor. Two-hop neighborhood information is obtained by each node by exchanging their neighbor lists periodically.

The use of one hop secrets can prevent external attackers from participating in the network (as packets not encrypted or authenticated with the group secret will be dropped). One-hop secrets can also be used to protect the RREQ relayed by a node from nodes not in its RDN, by encrypting the RREQ with the group secret.

The use of two-hop secrets can prevent attacks involving illegal lengthening or shortening of hop counts. Unfortunately, proactive maintenance of two-hop secrets can be expensive. The overheads become even more severe in highly dynamic networks where two hop topologies can change very frequently. For example, in a network where every node has (on an average) $r = 5$ neighbors, and if (on an average) the one hop topology of any node changes once every minute, the two hop topology will change once every 12 seconds (on an average). Thus maintaining a group secret at all times with each of the $\mathbb{O}(r^2)$ two-hop neighbors can demand significant bandwidth overheads.

C. Possible Modifications of SAODV

1) Secure RDN: A simple modification to SAODV that would address the most serious of its pitfalls (its susceptibility to external attackers) would be to require every intermediate node to authenticate itself using a digital signature. Even if such a signature is not carried forward (stripped at the next hop), external attackers can be eliminated.

A more effective alternative is to perhaps use the public-private key pair of every node to establish a group secret with all one-hop neighbors in the RDN (similar to the approach in [4]) and encrypt all broadcasts using the group secret. A node A could provide a group secret K_A to all its neighbors in the RDN. Every node proactively maintains its RDN when the topology changes (by changing the group secret). Such an approach can simultaneously keep external attackers out and ensure that nodes that are not in the RDN cannot forward the RREQ.

Perhaps a more efficient strategy to establish one-hop secrets is to employ a key predistribution scheme (KPS) that caters for pairwise authentication for individually encrypting the group secret to be conveyed. Motivated by rapidly shrinking cost of storage (even for mobile devices) some novel KPSs have been proposed recently [10], [11] that can resist even collusions of hundreds of thousands of nodes with with low computational requirements and less than 100 MB of storage per node. As flash cards supporting several GBs are already common mobile nodes could easily afford to use some of that storage for storing keys and / or authenticated public values.

Apart from facilitating an efficient mechanism for securely conveying one-hop secrets, such KDSs which cater for pairwise shared secrets can also be used for efficiently authenticating the RREP. Note that every node relaying the RREP will know the identity of the next hop, or next two hops if authentication was carried over by one hop in the RREQ.

2) Two-hop Authentication: At first sight it may appear that carrying over authentication to two-hops can prevent attacks involving illegal modifications to hop counts. Unfortunately this is not true. For example consider a scenario involving a path $\cdots A \rightarrow B \rightarrow C \rightarrow D$, where an RREQ reaches a malicious node C through the path $\cdots \rightarrow A \rightarrow B$. In such a scenario, A would have included an authentication for hop count i, and B, an authentication for hop-count $i+1$. Node C, with the knowledge $h^{i+1}(X)$ (supplied by B) could remove the authentication inserted by B and forward the RREQ with hop count $i+1$ (instead of $i+2$), and forward the authentication of A to the next node D (instead of stripping the authentication by A and forwarding authentication by B). Effectively, the malicious node C falsely portrays A as its neighbor.

In order for a downstream node D to determine that A cannot possibly be a one-hop neighbor of C, D requires authenticated knowledge of all one-hop neighbors of C. The use of two-hop group secrets and complete knowledge of two-hop topology makes this possible in [4]. Unfortunately, as mentioned earlier, maintaining two-hop groups and can require substantial overheads. The proposed extensions to SAODV employs an efficient *multi-source broadcast encryption* scheme, which while catering for two-hop authentication, does not require nodes to maintain two-hop groups.

[2] Nodes with whom bidirectional links exist

III. Multi Source Broadcast Encryption

Broadcast encryption (BE) ([12]) provides a means of establishing a shared secret between g privileged nodes, out of a universe of N nodes, where $g + r = N$, and the r nodes which are *not* provided with the secret are usually referred to a "revoked" nodes. Specifically, BE deals with cases where $r << N$.

BE is typically realized using some form of key predistribution, where a set of k secrets are distributed to each node in the universe of N nodes (before the system is deployed). The source of the broadcast then 1) chooses a broadcast secret K_b, 2) encrypts K_b using n keys $K_{e1} \cdots K_{en}$, and 3) transmits n values $K_{ei}(K_b), 1 \leq i \leq n$. The keys $K_{e1} \cdots K_{en}$ are chosen in such a way that 1) *none* of the r revoked nodes have access to *any* of the keys $K_{e1} \cdots K_{en}$, and 2) the remaining $N - r$ nodes in $\mathbb{G}_1 = \mathbb{G}_0 \setminus \mathbb{G}_R$ should have access to *at least one* of the secrets $K_{e1} \cdots K_{en}$, and thereby gain access to the secret K_b. Typically such a broadcast message takes the form

$$[(I_1 \cdots I_r) \parallel \{K_{ei}(K_b)\}], 1 \leq i \leq n \tag{1}$$

where $(I_1 \cdots I_r)$ explicitly specifies the list of nodes (by their unique IDs) who will not be provided access to the secret K_b.

Most popular BE schemes in the literature assume that the source of the broadcast is also the entity that distributed the secrets before deployment. It has been widely believed [13] until recently that facilitating *any* source to employ predistributed secrets to perform BE calls for asymmetric cryptographic primitives. Recently Ramkumar et al proposed multi-source BE schemes [14], [15] that cater for encrypted broadcasts by *any* node which has received predistributed secrets, using only inexpensive symmetric cryptographic primitives.

Apart from catering for BE by any node, the schemes proposed in [14], [15] are especially well suited for scenarios where 1) the total number of nodes in the network N is large; 2) the number of nodes to be revoked are small, and 3) the number of nodes g that actually *need* to gain access to the BE secret is small compared to the $N - r$. For example, a node in an ad hoc network (say with r neighbors and say $\mathbb{O}(r^2)$ nodes in the two hop region) can broadcast a message which indicates the list of its one-hop neighbors who are revoked, and include n encryptions of the broadcast secret in such a way that while none of the r one-hop neighboors can get access to the secret, all two hop neighbors (with a high probability) can get access to the secret (if such a message is forwarded by one hop neighbors). For a scenario where $r = 5$ the scheme in [14], [15] would require less than $n = 10$ encryptions of the broadcast secret.

A. BE Using A-RPS

In the "asymmetric" random preloaded subsets (A-RPS) scheme in [14], [15], a KDC chooses k secrets $K_1 \cdots K_k$ and a node A is provided with two sets of secrets, a set of k "encryption" secrets $\mathfrak{S}_A$ and a set of $m < k$ "decryption" secrets $\mathbb{S}_A$,

$$\begin{aligned} \mathbb{S}_A &= \{K_{A_1}, K_{A_2}, \ldots, K_{A_m}\} && (2)\\ \mathfrak{S}_A &= \{K_j^A = h(K_j \parallel A)\}, 1 \leq j \leq k. && (3) \end{aligned}$$

The indices of the decryption secrets assigned to any node is determined by a public one way function

$$F(A) = \{A_1 \cdots A_m\}, 1 \leq A_i \leq k \tag{4}$$

A broadcast secret chosen by A, say T_A, which is to be protected from r nodes $R_1 \cdots R_r$ is encrypted with some n encryption secrets (chosen from the k encryption secrets). As there may exist substantial freedom to choose the specific set of n secrets for this purpose, a public "rule" is enforced in choosing the optimal set of n secrets to be used by A for this purpose [14].

1) Revoking Neighbors: A node A with a set of neighbors $\mathbb{N}_A$ could choose a secret T_A and construct a BE message

$$\begin{aligned} \mathbb{B}_A &= [A \parallel \mathbb{N}_A \parallel \{K_{e_i}^A(T_A), 1 \leq i \leq n\} \parallel M_{T_A}^B] && (5)\\ M_{T_A}^B &= h(A, \mathbb{N}_A, \{K_{e_i}^A(T_A), 1 \leq i \leq n\}, T_A). && (6) \end{aligned}$$

Given the IDs of the source and all revoked nodes, any node which receives the message $\mathbb{B}_A$ can determine the indices e_i of each of the n encryption secrets used by the source (as this is determined by a fixed rule). Any node (say $X \notin \mathbb{N}_A$) which receives a packet with $\mathbb{B}_A$, can, 1) with a high probability decrypt T_A; 2) *verify* that none of the nodes in $\mathbb{N}_A$ could have access to the secret T_A, and thus 3) conclude that none of the nodes in the set $\mathbb{N}_A$ could have modified the packet $\mathbb{B}_A$ (using the HMAC $M_{T_A}^B$).

IV. Proposed Secure Route Discovery Protocol

For the proposed protocol we shall refer to as SAODV-2 (where 2 indicates the use of two-hop authentication) we assume 1) an offline KDC who distributed secrets / public values to every node to facilitate establishment of pairwise secrets between nodes; 2) an offline KDC who has distributed authentication and verification secrets of a MSBE scheme (like A-RPS) to every node; and 3) a public / private key pair and a certificate signed by an off-line certificate authority (CA) for every node (along with an authentic copy of the public key of the CA).

In SAODV-2 every node proactively maintains a secure RDN by providing a group secret to every node in the RDN. We shall represent the RDN secret of node A by K_A, which is randomly chosen by A and delivered to all nodes in its RDN by encrypting K_A with pairwise secrets. This RDN secret can also be used to cut off some nodes from their RDN if they are suspected of misbehavior. For example, if a node B with 4 nodes A, C, J and G in its RDN suspects C of malicious behavior, B can simply provide a new RDN secret K_B to its other neighbors A, G and J, thus cutting off C from its "logical" RDN (even though C is in the physical RDN of B).

A. Authentication Using Broadcast Secret

Apart from providing its RDN group secret K_A to every neighbor, a node A provides a BE message $\mathbb{B}_A$ with encrypted versions of a secret T_A. As all one-hop neighbors (nodes in the logical RDN) are revoked in the BE message, the one-hop neighbors will not gain access to the secret T_A. Thus a node B with logical RDN $\{A, G, J\}$ at some instant of time t will possess the RDN secrets K_A, K_G, K_H and the BE messages $\mathbb{B}_A, \mathbb{B}_H, \mathbb{B}_J$ respectively. Note that all such BE messages stored by B will indicate B as a member of *their* (nodes A, G and J) RDN. In other words, any node receiving the BE message $\mathbb{B}_H$ relayed by B can rest assured that B does not know T_H conveyed by the BE message.

Whenever node B requires to forward some broadcast by a neighbor, say an RREQ relayed by A, it checks if it has already had relayed the BE message $\mathbb{B}_A$. If B had not done so earlier, along with the RREQ it forwards, B also attaches the BE broadcast $\mathbb{B}_A$ of its predecessor A. Any downstream neighbor of B (say H) which receives $\mathbb{B}_A$ can (with a high probability) 1) extract T_A; 2) verify that B does not have access to T_A (by verifying that B is included in the neighbor list in $\mathbb{B}_A$); and 3) verify the integrity of $\mathbb{B}_A$ (by verifying the HMAC appended in $\mathbb{B}_A$).

Note that when H receives a BE message $\mathbb{B}_A$ relayed by B (that explicity revokes B) H is provided a guarantee that B is indeed a one hop neighbor of A and by extension (as B is a one hop neighbor of H), A is two-hops from H. It is important to see that while a node may never get to know the *entire* two-hop topology at any time, the use of BE can ensure that a *node securely recognizes its two hop neighbors, without having to trust the one-hop neighbor in between the two nodes*. Without *a priori* knowledge of the identity of the downstream nodes, node A can convey a secret T_A to any node that is *not* in the RDN (only nodes in the RDN are revoked). Node A can use this secret T_A to append a HMAC for verification by two-hop neighbors. This prevents an one-hop downstream neighbor of A from illegally modifying the RREQ before forwarding it onwards.

B. Route Discovery

Let us consider a scenario where S originates a route request packet for determining a route to D. The nonmutable fields of the RREQ, represented by $rreq$ consists of

$$\begin{aligned} rreq &= [S \parallel D \parallel seq_S \parallel seq_D \parallel h_c \parallel \tau_S \parallel s_{h_c} \\ &\quad \parallel \mathrm{SIG}_S \parallel \mathrm{CERT}_S] \end{aligned} \qquad (7)$$

where seq_D is the last known sequence number of D by S, τ_S is an absolute time after which RREQs for S will not be honored by intermediate nodes that cache the $rreq$, SIG_S is the digital signature appended by S (covering all quantities to the left) CERT_S the public key certificate of S.

As in SAODV [2], in SAODV-2 the source S chooses a *random* s_0 and computes h_c repeated hashes to arrive at $s_{h_c} = h^{h_c}(s_0)$. The broadcast by S which includes $rreq$ takes the form

$$\begin{aligned} S \rightarrow * &: [S \parallel K_S([rreq \parallel 0 \parallel s_0 \parallel M_0])] && (8) \\ s_0 &= h(rreq, K_{SD}), \quad s_1 = h(s_0) && (9) \\ M_0 &= h(\{rreq, 1, s_1\}, T_S) && (10) \end{aligned}$$

All fields in the message aired by S (except the ID S) are encrypted using the secret K_S so that only neighbors in the logical RDN of S can receive and process the RREQ. The value M_0 is for purposes of verification by two-hop neighbors of S (whose identities may not be known to S).

A neighbor A of S 1) decrypts the RREQ transmitted by A, 2) increases the hop count field to 1, and broadcasts

$$\begin{aligned} A \rightarrow * &: [A \parallel K_A([rreq \parallel 1 \parallel s_1 \parallel (M_0 \parallel S) \parallel M_1])] \\ M_1 &= h(\{rreq, 2, s_2\}, T_A), s_2 = h(s_1) \end{aligned} \qquad (11)$$

As mentioned earlier, if A had not relayed the current BE message $\mathbb{B}_S$ (which changes whenever the RDN of S changes) earlier, A also includes the message $\mathbb{B}_S$ along with the RREQ. Such messages will also be encrypted with A's one hop secret K_A.

A node B at the next hop decrypts the broadcast by A (using A's RDN secret K_A). If B has not handled the BE message $\mathbb{B}_S$, it extracts the shared secret T_S and verifies the HMAC M_0 appended by S. Note that in verifying M_0 node B is assured of the 1) integrity of the RREQ and that 2) A sent a valid $s_1 = h(s_0)$. On successful verification B strips M_0 and adds a HMAC M_2 for verification by its two hop downstream neighbors. Thus the broadcast by B takes the form

$$\begin{aligned} B \rightarrow * &: [B \parallel K_B([rreq \parallel 2 \parallel s_2 \parallel (M_1 \parallel A) \parallel M_2])] \\ M_2 &= h(\{rreq, 3, s_3\}, T_B). \end{aligned} \qquad (12)$$

Every intermediate node also includes the identity of the previous hop to facilitate the next hop to verify two-hop authentication appended. Intermediate nodes cache 1) RREQs they have forwarded and 2) note down the identities of the two[3] predecessor nodes in the RREQ

When the RREQ reaches the destination indicating a hop count of j and a value s_j, the destination can verify independently that s_j is consistent with the commitment s_{h_c} signed by the source and the hop count j indicated in the RREQ. Let us assume that the RREQ reached the destination through the path $\cdots J, L, M, N$. Note that the destination will be aware of the IDs of the two immediate predecessors (M and N) in the RREQ as the destination will verify the HMAC appended by node M.

1) Route Response: The destination invokes an RREP

$$\begin{aligned} rrep &= [S \parallel seq_S \parallel D \parallel seq'_D \parallel h'_c \parallel \tau_D \\ &\quad \parallel d_{h'_c} \parallel \mathrm{SIG}_D \parallel \mathrm{CERT}_D] \end{aligned} \qquad (13)$$

where

1) seq'_D is a fresher sequence number of D;
2) $h'_c = j$ (the hop count indicated in the RREQ);

[3] Only one in the case of one-hop neighbors of the source.

3) $d_{h'_c} = h^{h'_c}(d_0)$ where d_0 is a randomly chosen value by the destination; and
4) τ_D (for use by intermediate nodes responding to RREQs bound for D in the future).

The destination also appends a HMAC M_{DM} based on the secret K_{DM} it shares with the node M two-hops away and unicasts to its neighbor N

$$\begin{aligned} RREP_0 &= [D \parallel K_D([rrep \parallel 0 \parallel d_0 \parallel M_{DM}])] \quad (14) \\ M_{DM} &= h(\{rrep, 1, d_1\}, K_{DM}) \quad (15) \end{aligned}$$

The RREPs unicasted by nodes N and M along the reverse paths will now take the form

$$\begin{aligned} RREP_1 &= [N \parallel K_N([rrep \parallel 1 \parallel d_1 \parallel M_{DM} \parallel M_{NL}])] \\ M_{NL} &= h(\{rrep, 2, d_2\}, K_{NL}) \quad (16) \\ RREP_2 &= [N \parallel K_N([rrep \parallel 2 \parallel d_2 \parallel M_{DM} \parallel M_{MJ}])] \\ M_{MJ} &= h(\{rrep, 3, d_3\}, K_{MJ}) \quad (17) \end{aligned}$$

Thus the RREP packets are also encrypted in transit with the one-hop group secret, and authenticated using a HMAC to two-hop nodes. Note that RREPs can be efficiently authenticated in the reverse path as nodes already know the identities of the nodes two hops away.

2) RREP by Intermediate Nodes: Any intermediate node, say C, receiving an RREQ from node S bound for a destination D can raise an RREP if

1) it finds an RREQ or RREP from D in its cache with time τ_D greater that the current time
2) the last known sequence number seq_D indicated by the RREQ source is lower than the sequence number of D in the cached RREQ.

For example, consider a scenario where a node B which is i hops away from a node D and node B has in its cache a signed $rreq$ or $rrep$ from node D. Node B will also have access to the value d_i (the commitment for which can be found in the $rrep$ / $rreq$ and signed by D). When B receives an RREQ for D from some node V, B can invoke an RREP in which the $rrep$ field will take the form

$$rrep' = [V \parallel seq_V \parallel D \parallel RR_D] \quad (18)$$

where RR_D is *either* the $rreq$ or $rrep$ from D (in its cache). Once again note that node B will have knowledge of the last two hops in the RREQ path from V to B. Thus if the RREQ reaches B through X and Y the RREP by node B takes the form

$$\begin{aligned} RREP_0 &= [B \parallel K_B([rrep' \parallel i \parallel d_i \parallel M_{BX}])] \quad (19) \\ M_{BX} &= h(\{rrep', i+1, d_{i+1}\}, K_{BX}). \quad (20) \end{aligned}$$

As earlier intermediate nodes between B and V unicast the RREP exactly as for the case of RREP instantiated by the destination (Eqs (14) - (17)).

C. Security Analysis

Verifying the integrity of any route obtained from any flooding based route discovery protocol calls for the ability to 1) verify that no node can insert itself in the path (or forward an RREQ) without a valid authentication 2) verify that no value inserted by a node can be removed by a downstream node and 3) ensure that the reverse path also exists (or no one-way links).

Node insertion attacks are prevented by mandating authentication from every participating node. Every node authenticates itself to its one hop neighbors (using the group secret) and two-hop neighbors using the BE secret. Thus unless two nodes collude, nodes cannot be inserted illegally.

Prevention of node deletion attacks between any two end points calls for a shared secret between the end points. In other words, two nodes X and Y separated by any number of nodes can recognize node deletion attacks if X and Y share a secret. Thus while simply carrying over authentication does not prevent node deletion attacks, the ability to establish a secret with two-hop downstream neighbors provides the assurance that one-hop path between the two nodes cannot be modified.

1) One-Hop Secret: Note that use of RDN secret (or one-hop secrets) to encrypt all transmissions by any node, apart from 1) preventing the use of one-way links, 2) keeping external attackers out of the network, also 3) protects the per-hop hash value in any RREQ or RREP received by a node.

2) Per-hop Hashing: Irrespective of the whether a node receives a per-hop hash value of some node - 1) in a RREQ or 2) an RREP packet or even 3) an RREQ / RREP packet present *inside* an RREP packet when RREP is invoked by an intermediate node - a node i hops away from some node D will only have access to the value d_{i-1}.

While per-hop hashing is strictly not required (as authentication with two-hop secrets prevents attackers from both extending and shortening paths), it provides an independent confirmation for every intermediate node regarding the number of hops the RREQ or RREP have traversed. In other words, even nodes that have "some how" gained access to the BE secret of a node in the RDN cannot *shorten* the path. In other words, under such an eventuality, SAODV-2 offers the same protection as SAODV (except that SAODV-2 still keeps external attackers out). Another important reason for including this value is that the overhead required for computing / appending the per-hop hash value is trivial.

3) Broadcast Secret: The broadcast secret is essentially a secret established between a node and a predecessor two hops away, which prevents both insertion and deletion attacks as long as no two nodes collude together.

It is also important to see that SAODV-2 does *not* assume that nodes will advertise their RDN correctly in their BE messages. For example B could indicate in its message $\mathbb{B}_B$ a fictitious neighbor U. However, any packet that B claims to have passed through U *should* be supported with a BE message $\mathbb{B}_U$ which revokes B and is authenticated using a secret that B *cannot* have access to. If B cannot produce such a message, B cannot forward any RREQ indicating U as its predecessor.

It may appear at first sight that a node C which is two hops away from a node A (and had recently gained access to node A's BE secret T_A) can, by moving within one-hop of A, defeat the two-hop authentication process. However, note that when C enters the RDN of A, node A will change both the RDN secret K_A and the two-hop secret T_A. Thus the old T_A is of no use for C. Also note that a node C with the two-hop secret may also become a 3-hop node. However this has no effect on the security of the two-hop authentication process. A node which verifies two-hop authentication based on a BE secret will also ensure that the immediate predecessor is explicitly revoked in the BE message that conveys the BE secret.

All that any node needs to proactively keep track of is its one-hop neighbors. Changes in two-hop topology for instance 1) a two-hop node vanishing (powering off) or 2) two-hop node becoming a 3-hop node or 3) a two-hop node moving within one-hop has *no* effect on the security of the two-hop authentication process (as long as no two nodes collude).

The overheads required for two-hop authentication using BE is substantially smaller than the overheads required for maintaining consistent two-hop group secrets. For the latter case, for an average neighborhood size of r which changes (on an average) every t seconds, maintaining a consistent two hop group secret requires bandwidth overheads at a rate $\mathbb{O}(r^2t^2)$ as there will be $\mathbb{O}(r^2)$ nodes at a 2-hop distance and the two-hop topology changes at a rate proportional to $\frac{1}{t^2}$. Furthermore, consistent maintenance of two-hop secrets are required even during "quiet" periods where a node may not relay any RREQ.

On the other hand, when BE is used for two-hop authentication, message exchanges between neighboring nodes occur only when the RDN changes, calling for a rate of $\mathbb{O}(rt)$ instead of $\mathbb{O}(r^2t^2)$. The relay of BE messages to two hop neighbors occurs only if 1) the source of the BE message forwards an RREQ and 2) if the RDN of the source of the BE message has changed.

4) RREP Authentication: SAODV-2 does *not* employ the two-hop BE secrets for authentication of RREP. The primary advantage of BE is that it makes it possible for a node to convey a secret to some nodes that are two hops away even while the conveyor (source of the BE message) has no *a priori* knowledge of the identities of the nodes that are two hops away. This is indeed the situation during RREQ propagation. In the reverse path (RREP) however nodes *do* have *a priori* knowledge of the next two nodes in the path ahead (this information is gained from the RREQ). Thus there is no need to employ a weak group secret for authentication as a stronger form of authentication can be realized using pairwise secrets.

V. CONCLUSIONS

We proposed a novel and efficient scheme for authentication of mutable fields in the RREQ packet of AODV, using two-hop secrets that can be established by just maintaining a consistent one-hop topology, and without the knowledge of the two-hop topology. We argued that the overheads required for maintaining a consistent one-hop topology information and a secure RDN is necessary in any case to 1) keep external attackers out, 2) prevent use of one-way links, 3) conceal the per-hop hash value from nodes not in the RDN and 4) the ability to keep nodes out of the logical RDN of any node (which may be necessary to deal with misbehaving nodes). Realization of the proposed two-hop authentication strategy only mandates nodes to exchange a broadcast encryption message with the nodes in the RDN. We pointed out the existence of an efficient multi-source BE scheme that is very useful for this purpose.

We then pointed out many pitfalls of a popular secure AODV protocol SAODV and proposed some modifications to overcome its limitations. The proposed modifications involved use of RDN secret and the use of multi-source BE for two-hop authentication.

REFERENCES

[1] Web Link, http://www.ietf.org/html.charters/manet-charter.html

[2] M.G.Zapata, N.Asokan, "Securing Adhoc routing protocols," WISE 2002 , Atlanta, Georgia.

[3] A.A.Pirzada, C.McDonald, "Secure Routing with AODV Protocol," Asia-Pacific Conference on Communications,Perth,Western Australia , October 2005.

[4] X.Du, Y.Wang, J.Ge, Y.Wang,"A Method for Security Enhancements in AODV Protocol,"In Proceedings of the 17th International Conference on Advanced Information Networking and Applications, AINA 2003.

[5] K.Sanzgiri, B.Dahill, B.Levine, C.Shields, E.M. Belding-Royer, "A Secure Routing Protocol for Ad Hoc Networks,"In proceedings of 2002 IEEE international Conference on Network Protocols(ICNP),November 2002.

[6] Y-C Hu, A Perrig, D B.Johnson, "Ariadne:A Secure On-Demand Routing Protocol for Ad Hoc Networks," The 8th ACM International Conference on Mobile Computing and Networking, Atlanta, Georgia, September 2002.

[7] P Papadimitratos, Z.J.Haas, "Secure Routing for Mobile Ad Hoc Networks," Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference(CNDS 2002), San Antonio, Texas,2002.

[8] J. Kim, G. Tsudik, "SRDP: Securing Route Discovery in DSR," IEEE Mobiquitous'05, July 2005.

[9] C Perkins, P Bhagvat, "Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers," ACM SIGCOMM Symposium on Communication, Architectures and Applications, 1994.

[10] M. Ramkumar, "Securing Ad Hoc Networks With "Asymmetric" Probabilistic Key Predistribution Schemes," the 7th IEEE IA Workshop, West Point, NY, June 2006.

[11] M. Ramkumar, "I-HARPS: An Efficient Key Predistribution Scheme for Mobile Computing Applications," IEEE Globecom, San Francisco, CA, Nov 2006.

[12] A. Fiat, M. Noar, "Broadcast Encryption," Lecture Notes in Computer Science, Advances in Cryptology, Springer-Verlag, **773**, pp 480–491, 1994.

[13] D. Noar, M. Noar, J. Lotspiech, "Revocation and Tracing Routines for Stateless Receivers," Lecture Notes in Computer Science, Advances in Cryptology, Springer-Verlag, **2139**, 2001.

[14] M. Ramkumar, "Broadcast Encryption with Probabilistic Key Distribution and Applications," *Journal of Computers*, June 2006.

[15] M. Ramkumar, N. Memon, "Secure collaborations over message boards", *Int. J. Security and Networks*, **1** (1), 2006.

[16] Y-C Hu, A. Perrig, D.B. Johnson, "Rushing Attacks and Defense in Wireless Ad Hoc Network Routing Protocols," WiSe 2003, San Diego, CA, September 2003.

[17] K.A. Sivakumar, M. Ramkumar, "On the effect of one-way links on route discovery in DSR," ICCCN-06, Arlington, VA, Oct 2006.