Securing the Destination-Sequenced Distance Vector Routing Protocol (S-DSDV) *

Tao Wan Evangelos Kranakis P.C. van Oorschot

{twan, kranakis, paulv}@scs.carleton.ca
School of Computer Science, Carleton University, Ottawa, Canada

Abstract. *A mobile ad hoc network (MANET) is formed by a group of mobile wireless nodes, each of which functions as a router and agrees to forward packets for others. Many routing protocols (e.g., AODV, DSDV, etc) have been proposed for MANETs. However, most assume that nodes are trustworthy and cooperative. Thus, they are vulnerable to a variety of attacks. We propose a secure routing protocol based on DSDV, namely S-DSDV, in which a well-behaved node can successfully detect a malicious routing update with any sequence number fraud (larger or smaller) and any distance fraud (shorter, same, or longer) provided no two nodes are in collusion. We compare security properties and efficiency of S-DSDV with superSEAD. Our analysis shows that S-DSDV-R, a variation of S-DSDV with a risk window similar to that of superSEAD, offers better security than superSEAD with less network overhead.*

Keywords: DSDV, Routing Security, Wireless Security, Security Analysis

1 Introduction

A mobile ad hoc network (MANET) is formed by a group of wireless nodes, each of which performs routing functions and forwards packets for others. No fixed infrastructure (i.e., access point) is required, and wireless nodes are free to move around. A fixed infrastructure can be expensive, time consuming, or impractical. Another advantage of MANETs is the expansion of communication distance. In an infrastructure wireless network, nodes are restricted to move within the transmission range of access points. MANETs relax this restriction by cooperative routing protocols where every node forwards packets for every other node in the network. Potential applications of MANETs include military battle field, emergency rescue, campus networking, etc.

MANETs face all the security threats of wireline network routing infrastructures, as well as new threats due to the fact that mobile nodes have constrained resources and lack physical protection. One critical threat faced by most routing protocols is that a single misbehaving router may completely disrupt routing operations by spreading fraudulent routing information since a trustworthy and cooperative environment is often assumed. Consequences include, but are not limited to: 1) packets may not be able to reach their ultimate destinations; 2) packets may be routed to their ultimate destinations over non-optimal routes; 3) packets may be routed over a route under the control of an adversary.

Many mechanisms [21, 20, 1, 6, 19, 17] have been proposed for securing routing protocols by providing security services, e.g., entity authentication and data integrity, or

* Version: August 03, 2004.

by detecting forwarding level misbehaviors [11, 9]. However, most do not validate the factual correctness of routing updates. One notable protocol is superSEAD proposed by Hu et al. [7, 8]. SuperSEAD is based on the Destination-Sequenced Distance Vector (DSDV) routing protocol [13], and uses efficient cryptographic mechanisms, including one-way hash chains and authentication trees, for authenticating sequence numbers and distances of advertised routes. SuperSEAD can prevent a misbehaving node from advertising a route with 1) a sequence number larger than the one it received most recently (*larger sequence number fraud*); 2) a distance shorter than the one it received most recently (*shorter distance fraud*); and 3) a distance the same as the one it received most recently (*same distance fraud*). However, superSEAD does not prevent a misbehaving node from advertising a route with a) a sequence number smaller than any one it has received (*smaller sequence number fraud*); or b) a distance longer than any one it has received (*longer distance fraud*). Another disadvantage is that it assumes the cost of a network link is one hop, limiting its applicability. For example, it may not applicable to a DV which uses network bandwidth as a parameter for computing cost metrics.

1.1 Problems Addressed and Results

Smaller sequence number and longer distance frauds clearly violate the routing protocol specifications, and can be used for non-benevolent purposes (e.g., selfishness). Although the consequences of these frauds are often viewed as less serious than those of larger sequence number fraud or shorter distance fraud, we believe they still need to be addressed for many reasons, including: 1) they can be used by selfish nodes to avoid forwarding traffic, thus detecting these frauds would significantly reduce the means of being selfish; 2) it is desirable to detect any violation of protocol specifications even though its consequences may remain unclear or the probability of such violation seems low. Past experience has shown that today's naive security vulnerabilities can often be exploited to launch serious attacks and to cause dramatic damages in the future. For example, a vulnerability of TCP sequence number prediction was discussed as early as 1989 [3], but was widely thought to be very difficult to exploit given the extremely low probability (2^{-32}) of guessing a correct sequence number. It did not attract much attention until April 2004 when a technique was discovered which takes less time to predict an acceptable TCP sequence number.

In this paper, we propose the use of *consistency checks* to detect sequence number frauds and distance frauds in DSDV. Similar ideas [18] have been used for securing RIP [10]. Our protocol, namely S-DSDV, has the following security properties, provided no two nodes are in collusion: 1) detection of any distance fraud (longer, same, or shorter); 2) detection of both larger and smaller sequence number fraud. One notable feature of S-DSDV is that a misbehaving node surrounded by well-behaved nodes can be contained. Thus, misinformation can be stopped in the first place before it spreads into a network. Our efficiency analysis shows that S-DSDV-R, a variation of S-DSDV with a risk window (§3.4) similar to that of superSEAD, offers better security than superSEAD with less network overhead.

The sequel is organized as follows. Section 2 provides background information of distance vector routing protocols and DSDV. Section 3 presents overview and security analysis of SEAD. Relevant threats are discussed in Section 4. S-DSDV is presented

in Section 5 and analyzed in Section 6. Efficiency of S-DSDV is compared with super-SEAD by analysis and simulation in Section 7. We conclude in Section 8.

2 Background

In this section, we provide background information for simple distance vector routing protocols and DSDV [13]. $G = (V, E)$ denotes a network where V is a set of nodes and E is a set of links. A distance vector route r may consist of some or all of the following fields: dst - a destination node; seq - a sequence number; cst - a cost metric or distance; nhp - a next hop node; aut - an authentication value. For example, $r_u(w) = (w, seq(u,w), cst(u,w), nhp(u,w))$ denotes a route from u to w, where $seq(u,w)$, $cst(u,w)$, and $nhp(u,w)$ denote the sequence number, the cost, and the next hop of $r_u(w)$ respectively. Without ambiguity, we also use (w, seq, cst), (w, seq, cst, nhp), or (w, seq_u, cst_u, nhp_u) to denote $r_u(w)$.

2.1 Distance Vector Routing Protocols

In a traditional DV algorithm, each node $v_i \in V$ maintains a cost metric or a distance for each destination node v_j in a network. Let $d^t(v_i, v_j)$ be the distance from v_i to v_j at time t. Initially or at time 0,

$$d^0(v_i, v_j) = \begin{cases} 0 & \text{if } v_i = v_j \\ \infty & \text{if } v_i \neq v_j \end{cases}$$

Suppose at time 1, each node v_i learns all of its direct neighbors (denoted by $N(v_i)$) by some mechanism, e.g., receiving a special message from v_j may confirm v_j as a direct neighbor. Suppose each node v_i also knows the distance to each of its direct neighbors $v_j \in N(v_i)$, which can be the cost of the edge linking v_i and v_j, $c(v_i, v_j)$. At time 1, node v_i's routing table can be detailed as:

$$d^1(v_i, v_j) = \begin{cases} 0 & \text{if } v_i = v_j \\ c(v_i, v_j) & \text{if } v_j \in N(v_i) \\ \infty & \text{if } v_i \neq v_j \text{ and } v_i \notin N(v_i) \end{cases}$$

Each node broadcasts its routing table to all of its direct neighbors periodically or when a distance changes. At time t, v_i receives routing updates from each of its direct neighbors, and updates the distance to v_k in its routing table with the shortest of all known distances to v_k. Thus, at time $t+1$,

$$d^{t+1}(v_i, v_k) = \min_{v_j \in N(v_i)} \{d^t(v_j, v_k) + c(v_i, v_j)\}$$

The advantages of DV routing protocols include: simplicity, low storage requirement, and ease of implementation. However, they are subject to short or long term routing loops. Routing loops are primarily caused by the fact that selection of next hops is made in a distributed fashion based on partial and possibly stale information. Routing loops can be manifested during the propagation of routing updates by the problem of count-to-infinity [10]. To mitigate this problem, several mechanisms can be used: 1) limiting the maximum network diameter to k (*limited network boundary*), thus, the problem of count-to-infinity becomes count-to-k; 2) not advertising a route back to the

node this route is learned from (*split-horizon*); 3) advertising an infinite route back to the node this route is learned from (*split-horizon with poisoned reverse*).

2.2 DSDV

DSDV [13] is a routing protocol based on a DV approach, specifically designed for MANETs. DSDV solves the problem of routing loops and count-to-infinity by associating each route entry with a sequence number indicating its freshness. The split-horizon mechanism is not applicable to MANETs due to their broadcast nature. In a wireline network, a node can decide over which link (or to which node) a routing update will be sent. However, in a wireless ad hoc network, a routing update is transmitted by broadcast and can be received by any wireless node within the transmission range. Thus, it is impossible to selectively decide which nodes should or will receive a routing update.

In DSDV, a sequence number is linked to a destination node, and usually is originated by that node (the owner). The only case that a non-owner node updates a sequence number of a route is when it detects a link break on that route. An owner node always uses even-numbers as sequence numbers, and a non-owner node always uses odd-numbers. With the addition of sequence numbers, routes for the same destination are selected based on the following rules: 1) a route with a newer sequence number is preferred; 2) in the case that two routes have a same sequence number, the one with a better cost metric is preferred.

2.3 Security Threats to DSDV

DSDV guarantees all routes are loop free. However, it assumes that all nodes are trustworthy and cooperative. Thus, a single misbehaving node may be able to completely disrupt the routing operation of a whole network. We focus on two serious threats: the manipulation of sequence numbers and the manipulation of cost metrics. Specifically, a misbehaving node can poison other nodes' routing tables or affect routing operations by advertising routes with fraudulent sequence numbers or cost metrics.

To protect a routing update message against malicious modification, public key based digital signatures may be helpful. For example, v_i sends to v_j a routing update signed with v_i's private key. v_j can verify the authenticity of the routing update using v_i's public key. However, digital signatures cannot prevent a malicious entity with legitimate keying materials from advertising false information (e.g., false sequence numbers or distances). In other words, *message authentication cannot guarantee the factual correctness of a routing update*. For example, when v_i advertises to v_j a route for v_d with a distance of 2, v_j is supposed to re-advertise that route with a distance of 3 if it is the best route to v_i known by v_j. However, v_j could advertise that route with any distance value without being detected by a message authentication mechanism.

3 SEAD Review

Hu et al. [7, 8] made a first attempt to authenticate the factual correctness of routing updates using one-way hash chains. Their proposal, based on DSDV and called SEAD [7], can prevent a malicious node from increasing a sequence number or decreasing a distance of an advertised route. In the above example, v_j cannot successfully re-advertise

the route with a distance shorter than 2. However, SEAD cannot prevent v_j from advertising a distance of 2 or longer (e.g., 4). In SuperSEAD [8], they proposed to use combinations of one-way hash chains and authentication trees to force a node to increase the distance of an advertised route when it re-advertises that routing update. In the above example, v_j cannot advertise a distance of 2. However, v_j is free to advertise a distance longer than 3.

We review SEAD in the remainder of this section. Due to space limitations, we omit description of SuperSEAD since it involves complex usage of authentication trees. We give a brief introduction of one-way hash chains, then provide an overview of SEAD, including its assumptions, protocol details, security properties, and some limitations.

3.1 One-Way Hash Chains

A one way hash function, $h()$, is a function such that for each input x it is easy to compute $y = h(x)$, but given y and $h()$ it is computationally infeasible to compute x such that $y = h(x)$ [12]. A one way hash chain on x of a length n, $hc(x, n)$, can be constructed by applying $h()$ on a seed value x iteratively n times, i.e., $h^i(x) = h(h^{i-1}(x))$ for $i \geq 2$. Thus, $hc(x, n) = (h(x), h^2(x), \ldots, h^n(x))$. One property of one way hash chains is that given $h^i(x), h^j(x) \in hc(x, n)$ and $i < j$, it is easy to compute $h^j(x)$ from $h^i(x)$, i.e., $h^j(x) = h^{j-i}(h^i(x))$, but it is computationally infeasible to compute $h^i(x)$ from $h^j(x)$.

3.2 Assumptions

As with virtually all other secure routing protocols, SEAD requires cryptographic secrets for entity and message authentication. Public key infrastructure or pair-wise shared keys can meet such requirement. Other key establishment mechanisms can also be used. For simplicity, we assume that each node (v_i) has a pair of public key (V_{v_i}) and private key (S_{v_i}). Each node's public key is certified by a central authority trusted by every node in the network. To minimize computational overhead, every node also establishes a different symmetric secret key shared with every other node in the network. A secret key shared between v_i and v_j is denoted $K_{v_i v_j}$.

The network *diameter*, k, is the maximum distance between any two nodes in the network. Given a network $G = (V, E)$, $k = max\{d(u, v) | u, v \in V\}$. It would be ideal if a routing protocol can scale to any network without boundary limitation. However, a DV routing protocol is usually used in a small or medium size network. Thus, it is realistic for a DV routing protocol to assume a maximum network diameter k_m (e.g., $k_m = 15$ in RIP [10]). Nodes located k_m hops away are treated as unreachable.

3.3 Review of SEAD Protocol Details

SEAD authenticates the sequence number and the distance of a route with an authentication value which is an element of a hash chain. To advertise a route $r_{v_i}(v_d, seq, cst)$, v_i needs to include an authentication value $aut(r_{v_i})$ to allow a recipient to verify the correctness of r_{v_i}. The following summarizes how SEAD works:

1. Let s_m be the maximum sequence number. $\forall v_i \in V$, v_i constructs a hash chain from a secret x_i, $hc_{v_i}(x_i, n+1) = (h^1(x_i), h^2(x_i), \ldots, h^{n+1}(x_i))$. We assume $n = s_m \cdot k_m$ for the sake of simplicity. Arrange $hc_{v_i}(x_i, n+1)$, or simply hc_{v_i},

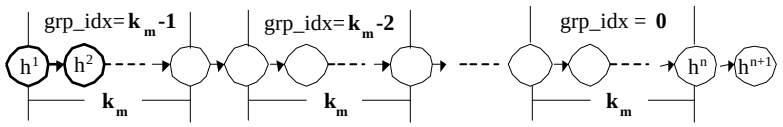

Fig. 1. A hash chain is arranged into groups of k_m elements.

into s_m groups of k_m elements. The last element $h^{n+1}(x_i)$ is not in any group and is referred as the *anchor* of hc_{v_i}. Each group is assigned an integer in the range $[0, k_m - 1]$ as its index. Groups are numbered from right to left (Fig. 1). The hash elements within a group are numbered from left to right starting from 0 to $k_m - 1$. This way, each hash element $h^j(x_i)$ can be uniquely located within hc_{v_i} by two numbers a, b, where a is the index of the group which $h^j(x_i)$ is in and b is the index of the element within the group. We use $hc_{v_i}[a, b]$ to represent $h^j(x_i)$, where $j = (s_m - a) \cdot k_m + b + 1$.

2. $\forall v_i \in V$, v_i makes $h^{n+1}(x_i)$ accessible to every other node in the network. Many methods can be used. For example, v_i can publish $h^{n+1}(x_i)$ in a central directory, signing it with v_i's private key. Another method is to broadcast to the whole network $h^{n+1}(x_i)$ along with v_i's digital signature. The result is that every node in the network has a copy of $h^{n+1}(x_i)$ and can trust that it is the correct anchor value of a hash chain constructed by v_i.
3. $\forall v_i \in V$, v_i advertises to its neighbors a route r_{v_i} for v_k with a distance of d and a sequence number of s, $r_{v_i} = (v_k, s, d)$. To support r_{v_i}, v_i includes an authentication value $aut = hc_{v_k}[s, d]$ with r_{v_i}.

$$v_i \rightarrow N(v_i) : \ r_{v_i}(v_k, s, d, aut), \ aut = \begin{cases} hc_{v_i}[s, 0] & \text{if } v_k = v_i \\ hc_{v_k}[s, d] & \text{if } v_k \neq v_i \end{cases}$$

4. Upon receiving an advertised route $r_{v_i}(v_k, s, d, aut)$, v_j validates d and s using the one-way hash chain. We know that aut should be $hc_{v_k}[s, d]$, or $h_{v_k}^{(s_m - s) \cdot k_m + d + 1}(x_k)$. Given the anchor of $hc_{v_k} = h_{v_k}^{n+1}(x_k) = h_{v_k}^{s_m \cdot k_m + 1}$, it is easy to confirm if $aut = hc_{v_k}[s, d]$ by applying $h()$ on aut for x times, where $x = (s_m \cdot k_m + 1) - [(s_m - s) \cdot k_m + d + 1] = s \cdot k_m - d$. If $aut = hc_{v_k}[s, d]$, then $r_{v_i}(v_k, s, d, aut)$ is treated as valid. Otherwise, invalid. In the former case, r_{v_i} is used to update the existing route in v_j's routing table for v_k, let's say $r_{v_j}(v_k, s', d', aut')$ if 1) $s > s'$ or 2) $s = s'$ and $d < d'$. In either case, d', s' and aut' are replaced with $d + 1$, s and $h(aut)$ respectively.

3.4 Security Analysis of SEAD

SEAD has a number of desirable security properties (Table 1):

1. Data origin authentication and data integrity.
2. Sequence number authentication. Provided no two nodes are in collusion, a bad node cannot corrupt another node's routing table by advertising a route with a sequence number greater than the latest one originated by the destination of that route.

3. Cost metric authentication. Provided no two nodes are in collusion, a bad node cannot corrupt another node's routing table by advertising a route with a distance shorter than the one it learns from one of its neighbors.
4. Partial resilience to collusion. Given a group of colluding nodes, the shortest distance they can claim to a destination x without being detected is the shortest distance from any node in the colluding group to x. For example, if u, v are in collusion, and u, v are 3 and 5 hops away from x respectively, then the shortest distance to x which u and v can claim is 3 hops. Thus, we say that SEAD partially resists collusion since colluding nodes are unable to arbitrarily falsify a distance.

Security Property		SEAD	superSEAD	S-DSDV
Data Integrity		✓	✓	✓
Data Origin Authentication		✓	✓	✓
Destination Authentication		✓	✓	✓
Sequence Number Authentication	larger	✓	✓	✓
	smaller	×	×	✓
Cost Metric Authentication	longer	×	×	✓
	same	×	✓	✓
	shorter	✓	✓	✓
Resistance to 2-node collusion		⋄	⋄	×

Table 1. Security Comparison of SEAD, superSEAD, and S-DSDV:
× - not supported; ⋄ - partially supported; ✓ - fully supported;

Despite its distinguishable security properties, SEAD has some limitations.

1. *Vulnerable to longer distance fraud*. A misbehaving node can advertise a route with a distance longer than the actual distance of that route without being detected. For example, a node i located k hops away from j can successfully advertise a route for j with a distance $d > k$. This is possible because i has received $h^{k-1}()$ and can compute it forward to obtain $h^d()$ to authenticate distance d.
2. *Vulnerable to lower sequence number fraud*. A misbehaving node i can advertise a sequence number lower than the one it receives. Thus, i may be able to advertise a shorter distance route by lowering its sequence number.
3. *A risk window*. SEAD has a risk window of p_1, which is the interval of periodic routing updates. For example, a node i which had been k hops away from j can still claim that distance when it actually has moved further away from j since i has the authentication value $h^k()$ to support its claim. Such a claim would continue being valid until a victim receives a route for j from other nodes with a newer sequence number. Although such a risk window is usually short (e.g., 15 seconds in SEAD), it is still desirable to minimize it.

4 Threats to Routing Protocols

A routing protocol faces many threats. In this section, we discuss such threats and identify those addressed by our protocol.

4.1 Threat Targets

The primary objective of the network layer is to provide routing functionality to allow non-directly connected nodes to communicate with each other. This implies two fundamental functions for a router: (1) Establishing valid routes (usually stored in a routing table) to destinations in a network. Automatic mechanisms for building and updating routing tables are often referred to as *route propagation mechanisms* or *routing protocols*. (2) Routing datagrams to next hops leading to their ultimate destinations. Such function is often referred to as *routing algorithms*. Example routing strategies include, but are not limited to: a) routing datagrams to a default gateway; b) routing datagrams over shortest paths; c) routing datagrams equally over multiple paths; d) policy routing; e) stochastic routing.

Although these two functions are equally important and both deserve attention, this paper only considers threats against automatic route propagation mechanisms, specifically, DSDV. A routing protocol is usually built upon other protocols (e.g., IP, TCP, or UDP). Thus, it is vulnerable to all the threats against its underlying protocols (e.g., IP spoofing). In this paper, we do not consider threats against underlying protocols. However, some of these threats can be mitigated by proposed cryptographic mechanisms.

4.2 Threat Sources

In a wireline network, threats can be from a network node or a network link (i.e., under the control of an attacker). Attacks from a controlled link include modification, deletion, insertion, or replay of routing update messages. In a MANET, attacks from network links are less interesting due to the broadcast nature of wireless networks. It appears difficult, if not impossible, for an attack to modify or delete a message (m), i.e., to stop the neighbors of m's originator from receiving untampered m. However, insertion and replay are more feasible. For simplicity, we model a compromised network link as an adversary node. A misbehaving node could be an *insider* (i.e., a compromised node with legitimate cryptographic credentials), or an *outsider* (i.e., a node brought to the network by an attacker without any legitimate cryptographic credentials).

Generic Threats	Addressed by S-DSDV?
Deliberate Exposure	×
Sniffing	×
Traffic Analysis	×
Byzantine Failures	◇
Interference	◇
Overload	✓
Falsification by Originators	✓
Falsification by Forwarders	✓

Table 2. Routing Threats: × - no; ◇ - partially; ✓ - fully;

4.3 Generic Threats

Barbir, Murphy and Yang [2] identified a number of generic threats to routing protocols, including *Deliberate Exposure, Sniffing, Traffic Analysis, Interference, Overload, Spoofing, Falsification, and Byzantine Failures* (Table 2). We consider falsification as one of the most serious threats to DSDV due to the fact that each node builds its own routing table based on other nodes' routing tables. This implies that a single misbehaving node may be able to compromise the whole network by spreading falsified routing

updates. Our proposed S-DSDV can defeat this serious threat by containing a misbehaving node (i.e., by detecting and stopping misinformation from further spreading).

5 S-DSDV

In this section, we present the details of S-DSDV, which can prevent any distance fraud, including longer, same, or shorter, provided no two nodes are in collusion.

Cryptographic Assumptions. As any other secure routing protocol, S-DSDV requires cryptographic mechanisms for entity and message authentication. Any security mechanisms providing such security services can meet our requirements, e.g., pair-wise shared secret keys, public key infrastructure (PKI), etc. Thus, S-DSDV has similar cryptographic assumptions as SEAD (see §3.2) and S-AODV (requiring PKI). We assume that every node ($v_i \in V$) shares with every other node ($v_j \in V, i \neq j$) a different pair-wised secret key (k_{ij}). Combined with message authentication algorithms (e.g., HMAC-MD5), pair-wise shared keys provide entity and message authentication. Thus, all messages in S-DSDV are cryptographically protected. For example, when i sends a message m to j, i also sends to j the Message Authentication Code (MAC) of m generated using k_{ij}.

5.1 Route Classification

We classify routes $R_u = \{r_u\}$ advertised by node u into two categories (as defined by Definitions 1, 2): 1) those u is authoritative of (R_u^{auth}); and 2) those u is nonauthoritative of (R_u^{naut}). $R_u = R_u^{auth} \cup R_u^{naut}$.

Definition 1 *(**Authoritative Routes**). Let $r_u = (w, seq, cst)$. $r_u \in R_u^{auth}$ if 1) $w = u$ and $cst = 0$; or 2) $cst = \infty$.*

It is obvious that u is authoritative of r_u if r_u is a route for u itself with a distance of zero. We also say that u is authoritative of r_u if r_u is an unreachable route. This is because u has the authority to assert the unavailability of a route from u to any other node w even if there factually exists such a path between u and w. This is equivalent to the case that u implements a local route selection policy which filters out traffic to and from w. We believe that a routing protocol should provide such flexibility for improving security since u may have its own reasons to distrust w. BGP [14] is an example which allows for local routing policies. However, this feature should not be considered the same as malicious packet dropping [11, 9], wherein, a node promises to forward packets to another node (i.e., announcing reachable routes to that node) but fails to do so.

Definition 2 *(**Non-Authoritative Routes**). Let $r_u = (w, seq, cst)$. $r_u \in R_u^{naut}$ if $r_u \notin R_u^{auth}$, i.e., $w \neq u$ and $0 < cst < \infty$.*

If u advertises a reachable route r_u for another node w, we say that u is not authoritative of r_u since u must learn r_u from another node, i.e., the next hop from u to w along the route r_u.

5.2 Route Validation

When a node v receives a route r_u from u, v validates r_u based on the following rules.

Rule 1 *(***Validating Authoritative Routes***). If u is authoritative of r_u, a recipient node v validates the MAC of r_u. If it succeeds, v accepts r_u. Otherwise, v drops r_u.*

Since u is authoritative of r_u, v only needs to verify the data integrity of r_u, which includes data origin authentication [12]. If it succeeds, v accepts r_u since it in fact originates from u and has not been tampered with. Otherwise, r_u is ignored since it might have originated from a node impersonating u or have been tampered with.

Rule 2 *(***Validating Non-Authoritative Routes***). If u is nonauthoritative of r_u, a recipient node v verifies the MAC of r_u. If it succeeds, v additionally validates the consistency (per Definition 3) of r_u. If it succeeds, v accepts r_u; otherwise, drops r_u.*

Since u is nonauthoritative of r_u, v should not accept r_u right away even if the validation of data integrity succeeds. Instead, v should check the consistency with the node which r_u is learned from. Ideally, v should consult with the authority of r_u if it exists. Such an authority should have perfect knowledge of network topology and connectivity (i.e., it knows every route and its associated cost from every node to every other node in a network). Such authority may exist for a small static network. However, it does not exist in a dynamic MANET where nodes may move frequently. Thus, we propose that v should consult with the node which r_u is learned from, which should have partial authority of r_u. This method is analogous to the way human beings acquire their trust by corroborating information from multiple sources.

Definition 3 *(***Consistency***) Given a network $G = (V, E)$, let $u, v, w \in V$ and link $e(u, v) \in E$. Let $r_u(w) = (w, seq(u, w), cst(u, w))$ be directly computed from $r_v(w) = (w, seq(v, w), cst(v, w))$. We say that $r_u(w)$ and $r_v(w)$ are* **consistent** *if 1) $seq(u, w) = seq(v, w)$; and 2) $cst(u, w) = cst(u, v) + cst(v, w)$.*

From Definition 3, we know that r_u and r_v are consistent if r_u is directly computed from r_v following DSDV specifications: 1) the sequence number should not be changed; and 2) the cost metric of r_u should be the sum of the cost metrics of r_v and $e(u, v)$. To complete a consistency check, a node needs to consult another node 2 hops away. Thus, we require that the next hop of a route be advertised along with that route. For example, if u learns a route $r_u(w)$ from v, u should advertise $r_u(w) = (w, seq(u, w), cst(u, w), nhp(u, w))$, where $nhp(u, w) = v$. To check the consistency of $r_u(w)$, a node x sends a route request to v, asking for v's route entry for w, which is $r_v(w) = (w, seq(v, w), cst(v, w), nhp(v, w))$. In addition, x also asks v's route entry for u, which is $r_v(u) = (u, seq(v, u), cst(v, u), nhp(u, v))$. Assuming $cst(v, u) = cst(u, v)$, $cst(v, u)$ allows x to check the consistency of $cst(u, w)$ and $cst(v, w)$. $nhp(v, u)$ allows x to check if u is directly connected with v, i.e., if $nhp(v, u) = u$.

5.3 Protocol Summary

The following is a summary of how S-DSDV works:

1. $\forall u, w \in V$, u advertises $r_u = (w, seq, cst, nhp)$ for w. Note r_u is MAC-protected.
2. Upon receiving r_u from u, $x \in V$ validates the MAC of the message carrying r_u. If it fails, r_u is dropped. Otherwise, x further determines if u is authoritative of r_u (Definition 1). If yes, x accepts r_u. Otherwise, x checks the consistency of r_u with the next hop (nhp) (see Step 3). If it succeeds, r_u is accepted; otherwise, dropped.
3. Let $v = nhp$. x sends a route request to v (e.g., via u), asking $r_v(w)$ and $r_v(u)$. v sends back a route response of $r_v(w)$ and $r_v(u)$. Upon receiving them, x performs a consistency check of $r_u(w)$ and $r_v(w)$ according to Definition 3. Note u may modify x's route request and/or v's route response. However, such misbehavior will not go unnoticed since all message are MAC-protected.

6 Security Analysis of S-DSDV

In this section, we analyze security properties of S-DSDV. We hope that our security analysis methodology can lead to a common framework for analyzing and comparing different securing routing proposals.

Theorem 1 (Data Integrity) *In S-DSDV, data integrity is protected.*

Proof Outline. S-DSDV uses pair-wise shared keys with a message authentication code (MAC) to protect integrity of routing updates. A routing update message with an invalid MAC will be detected.

Remark. Data integrity can prevent unauthorized modification and insertion of routing updates. However, it cannot by itself prevent deletion or replay attacks. Thus it partially counters the threat of interference [2].

Theorem 2 (Data Origin Authentication) *In S-DSDV, data origin is authenticated.*

Proof Outline. S-DSDV uses pair-wise shared keys with a MAC to protect integrity of routing updates. Since every node shares a different key with every other node, a correct MAC of a message also indicates that the message originated from the only other party the recipient shares a secret key with. Thus, data origin is authenticated.

Remark. Data origin authentication can prevent node impersonation since any node not holding the key materials of x cannot originate messages using x as the source without being detected. It can also thwart the threat of falsification by originators [2].

Given a route update $r = (dst, seq, cst, nhp)$ in S-DSDV, the threat of falsification by forwarders can be instantiated as follows: 1) falsifying the destination dst, i.e., using a dst which is not authorized to be in the network; 2) falsifying the sequence number seq; 3) falsifying the cost metric cst; 4) falsifying the next hop nhp. The lemmas below show that S-DSDV can resist these threats.

During a consistency check, a malicious node might also try to create the impression that other nodes are providing incorrect information by: 1) providing false route responses; 2) not responding to route requests; or 3) not forwarding route requests/responses. Since these types of fraud (namely *disruption fraud*) will lead to consistency check failures, correct route updates advertised by well-behaved nodes may be

dropped. We view this as a good trade-off between security and efficiency since it might be desirable not to use a route involving a misbehaving node although we do not know exactly which node is misbehaving. For the sake of simplicity, we do not consider disruption fraud in the following security analysis since it will result in consistency check failures and will thus be detected.

Lemma 1 (Destination Authentication) *In S-DSDV, a route with a falsified destination will be detected.*

Proof Outline. Since S-DSDV assumes pair-wised shared secret keys, we know that $\forall u, v \in V$ and $u \neq v$, u shares a secret key with v. If a destination node (x) in r is falsified or illegitimate, then $\forall u \in V$, u does not share a secret with x. Thus, a falsified destination node x will be detected.

Lemma 2 (Sequence Number Authentication) *In S-DSDV, an advertised route r with a falsified sequence number will be detected provided there is at most one bad node in the network.*

Proof Outline. Let b the bad node in the network, advertising $r_b = (x, seq_b, cst_b, nhp)$ to all of its direct neighbors $N(b)$, where seq is falsified (i.e., it is different from the value b learns from nhp). Since there is at most one bad node (b) in the network, $\forall u \in V, u \neq b$, u is a good node. By assumption, each of b's direct neighbors is good, including nhp. Thus, $\forall v \in N(b), v \neq nhp$, v will check the consistency of seq with nhp. Since nhp is a good node, it will provide a correct sequence number which will be inconsistent with seq_b if seq_b is faked. Therefore, the statement is proved.

Lemma 3 (Cost Metric Authentication) *In S-DSDV, an advertised route r with a falsified cost metric will be detected if there is at most one bad node in the network.*

Proof Outline. Since a good node can uncover misinformation from a bad node by cross checking its consistency with a good node, a falsified cost metric always causes inconsistency, and thus will be detected (see proof for Lemma 2).

Lemma 4 (Next Hop Authentication) *In S-DSDV, an advertised route r with a falsified next hop will be detected if there is at most one bad node in the network.*

Proof Outline. Let b the bad node in the network, which advertises $r = (x, seq, cst, nhp)$. We say nhp is falsified if: 1) $nhp \notin V$; or 2) $nhp \notin N(b)$; or 3) $nhp \in N(b)$ but r is not learned from nhp. If $nhp \notin V$, it will be detected by MAC failure since a legitimate node does not share a secret key with nhp. If $nhp \notin N(b)$, nhp will report a node $a \neq b$ as its next hop to b. If r is not learned from nhp, nhp will report a route to x with a distance inconsistent with cst. Therefore, Lemma 4 is proved.

Theorem 3 (Routing Update Authentication) *In S-DSDV, a routing update with falsified information will be detected provided there is at most one bad node in a network.*

Proof Outline. A routing update R consists of a number of routes (r). Based on Lemmas 1, 2, 3, and 4, we know $\forall r \in R$, any falsified information in any of the four fields in r will be detected if there is at most one bad node in the network. Therefore, it follows that falsified information in any part of R will be detected.

Definition 4 (Collusion) *Let x be the node advertising a route r_x, y be the next hop node of r_x, and r_y be the route provided by y during a consistency check of r_x. Let $r_x \Leftrightarrow r_y$ denote r_x and r_y are consistent, and $r_x \nLeftrightarrow r_y$ denote r_x and r_y are inconsistent. x and y are in* **collusion** *if y intentionally provides a falsified r_y such that $r_y \Leftrightarrow r_x$.*

Theorem 4 (Authentication in Presence of Multiple Bad Nodes) *Let $G = (V, E)$ be a network with n nodes, and maximum diameter k_m. Let s_m be the maximum sequence number in S-DSDV. Suppose G has $b \geq 2$ bad nodes, no two of which are in collusion. Suppose attackers randomly choose false sequence numbers, cost metrics, and next hops from $[1, s_m], [1, k_m], V$ respectively. Then, S-DSDV will detect any falsified route in a routing update with probability at least $1 - \frac{b-1}{n(n-1)s_m k_m}$.*

Proof Outline. Let node x advertise a route $r_x(w) = (w, seq(x,w), cst(x,w), y)$. Let $r_y(w) = (w, seq(y,w), cst(y,w), nhp(y,w)), r_y(x) = (x, seq(y,x), cst(y,x), nhp(y,x))$ be advertised by node y during a consistency check of $r_x(w)$. Suppose x is bad. If y is good, then a falsified $r_x(w)$ is always inconsistent with $r_y(w)$. Thus, it is always detected. We look at the probability that 1) y is bad (not in collusion with x) and 2) $r_x(w) \Leftrightarrow r_y(w)$, which is the probability that a falsified $r_x(w)$ goes undetected (denoted by $p(detection\ failure)$). $r_x \Leftrightarrow r_y$ requires: 1) $seq(x,w) = seq(y,w)$; 2) $cst(x,w) = cst(y,w) + cst(y,x)$; and 3) $nhp(y,x) = x$. Assuming that sequence numbers, distances, and next hops are randomly chosen from their allowed spaces, then $p(seq(x,w) = seq(y,w)) = \frac{1}{s_m}$, $p(cst(x,w) = cst(y,w) + cst(y,x)) = \frac{1}{k_m}$, $p(nhp(y,x) = x) = \frac{1}{n}$. Since $p(y\ is\ bad) = \frac{b-1}{n-1}$, $p(detection\ failuire) = \frac{b-1}{n(n-1)s_m k_m}$. Thus, $p(detection\ success) = 1 - \frac{b-1}{n(n-1)s_m k_m}$. Note a smart attacker trying to avoid detection by using a sequence number which differs from a correct one by a limited amount can decrease $p(detection\ success)$.

7 Efficiency Analysis

We analyze routing overhead caused by S-DSDV (S-DSDV overhead) and compare it with that caused by DSDV, SEAD, and superSEAD.

7.1 Analysis Methodology

We adopt a method of using both analysis and simulation for comparing routing overhead. Analysis has the advantage that it is easy for others to verify our results. Simulation has the advantage of dealing with the implications of random events which are difficult to obtain by analysis.

To analyze routing overhead, we need to obtain the total number of routing updates generated by all nodes in a network during a period of T time units. In DSDV, there are two types of routing updates: 1) periodic routing updates; and 2) triggered routing updates. In theory, the total number of periodic routing updates (U_{pd}) can be calculated. However, the total number of triggered updates (U_{tg}) cannot be easily calculated since they are related to random events, i.e., broken links caused by node movement. In the absence of an analytic method for computing the number of broken links resulting from a node mobility pattern, we use simulation to obtain U_{tg}. We also use simulation to obtain U_{pd} since it is affected by U_{tg} in the DSDV implementation in NS-2 [4]. For simplicity, we use the following assumptions and notation:

1. DSDV, SEAD, and S-DSDV run over UDP and IP. A routing update message including IP and UDP headers larger than 1500 bytes is split into multiple messages.
2. Each triggered routing update consists of a single entry for a route involved in the triggering event. If there are multiple routes affected by that event, multiple triggered routing updates are generated.
3. A DSDV route entry consists of a destination (4-byte), a sequence number (4-byte), and a cost metric (2-byte). Thus, $L_{dsdv\_rt} = 10$ bytes.
4. A SEAD route entry consists of a DSDV route entry plus a field of length L_{hash} for holding an authentication value. In this paper, we assume $L_{hash} = 80$ bits (10 bytes). Thus, $L_{sead\_rt} = 20$ bytes.
5. A superSEAD route entry consists of a DSDV route entry plus (k+1) fields of length L_{hash} for authentication values, where $k = lg(n)$ ($lg \equiv log_2$). In this paper, $k = lg(64) = 6$. Thus, $L_{ssead\_rt} = L_{dsdv\_rt} + (k+1) \times L_{hash} = 80$ bytes.
6. An S-DSDV route entry consists of a DSDV route entry plus a 4-byte length field holding the identity of a next hop node. Thus, $L_{sdsdv\_rt} = L_{dsdv\_rt} + 4 = 14$ bytes.
7. An S-DSDV consistency check involves a route request and a response message; each message has an S-DSDV route entry (plus IP and UDP headers), and traverses two hops. Thus, routing overhead generated per consistency check is $O_{sdsdv\_pcc} = (L_{sdsdv\_rt} + L_{ip\_hdr} + L_{udp\_hdr}) \times 4 = 168$ bytes.

Notation	Description	Value
L_{udp_hdr}	length of a UDP header	8 bytes
L_{ip_hdr}	length of an IP header	20 bytes
L_{hash}	length of a hash from a hash function	10 bytes
L_{dsdv_rt}	length of a DSDV route entry	10 bytes
L_{sead_rt}	length of a SEAD route entry	20 bytes
L_{ssead_rt}	length of a SuperSEAD route entry	80 bytes
L_{sdsdv_rt}	length of an S-DSDV route entry	14 bytes
O_{dsdv_ppu}	DSDV overhead per periodic routing update	528 bytes
O_{dsdv_ptu}	DSDV overhead per triggered routing update	38 bytes
O_{sead_ppu}	SEAD overhead per periodic routing update	1028 bytes
O_{sead_ptu}	SEAD overhead per triggered routing update	48 bytes
O_{ssead_ppu}	superSEAD overhead per periodic routing update	4612 bytes
O_{ssead_ptu}	superSEAD overhead per triggered routing update	118 bytes
O_{sdsdv_ppu}	S-DSDV overhead per periodic routing update	728 bytes
O_{sdsdv_ptu}	S-DSDV overhead per triggered routing update	42 bytes
O_{sdsdv_pcc}	S-DSDV overhead per consistency check	168 bytes
U_{pd}	total number of periodic routing updates	$*$
U_{tg}	total number of triggered routing updates	$*$
U_{tc}	total number of S-DSDV consistency checks	$*$
U_{pc}	total number of S-DSDV periodic consistency checks	$*$
O_{dsdv}	total DSDV overhead	†
O_{sead}	total SEAD overhead	†
O_{ssead}	total superSEAD overhead	†
O_{sdsdv_r}	total S-DSDV-R overhead	†
O_{sdsdv}	total S-DSDV overhead	†

Table 3. Notation for Effi ciency Analysis ($*$ - obtained by simulation; † - dependent on $*$ values)

We expected that S-DSDV produces relatively high network overhead since it checks the consistency of a route whenever it is updated for sequence number, distance, or the next hop. Since the sequence number changes persistently, a large number of consistency checks are triggered. To reduce S-DSDV overhead, we introduce a variation of S-DSDV, namely S-DSDV-R, which checks the consistency of a route when it is first installed in a routing table. A timer is set for that route when a consistency check is performed for that route. In our simulation, the timer interval is the same as the routing update interval. A new consistency check is only performed for a route when its consistency check timer expires. One security vulnerability of S-DSDV-R is that a falsified route may go undetected during the interval between two consistency checks. This is similar to the risk window of SEAD and superSEAD (§3.4). We use the following equations to calculate network overhead of each protocol:

$$O_{dsdv} = O_{dsdv\_ppu} \cdot U_{pd} + O_{dsdv\_ptu} \cdot U_{tg} \quad (1)$$

$$O_{sead} = O_{sead\_ppu} \cdot U_{pd} + O_{sead\_ptu} \cdot U_{tg} \quad (2)$$

$$O_{ssead} = O_{ssead\_ppu} \cdot U_{pd} + O_{ssead\_ptu} \cdot U_{tg} \quad (3)$$

$$O_{sdsdv\_r} = O_{sdsdv\_ppu} \cdot U_{pd} + O_{sdsdv\_ptu} \cdot U_{tg} + O_{sdsdv\_pcc} \cdot U_{pc} \quad (4)$$

$$O_{sdsdv} = O_{sdsdv\_ppu} \cdot U_{pd} + O_{sdsdv\_ptu} \cdot U_{tg} + O_{sdsdv\_pcc} \cdot U_{tc} \quad (5)$$

7.2 Simulation Results

We used simulation to obtain U_{pd}, U_{tg}, U_{pc}, and U_{tc}. We simulated a network with $n = 50$ mobile nodes for T = 900 seconds. Different pause times represent different dynamics of a network topology. A pause time of 0 seconds represents a constantly changing network, while a pause time of 900 seconds represents a static network. Simulation results are illustrated by Fig. 2. We observed that S-DSDV produces higher network overhead than superSEAD due to a significant number of consistency checks; we view this as the price paid for improved security. S-DSDV-R significantly reduces the network overhead, and offers better security than superSEAD, while having a similar risk window as superSEAD. However, the S-DSDV-R risk window can be managed by adjusting the value of the consistency check timer. Overall, we think S-DSDV-R provides a desirable balance between security and efficiency.

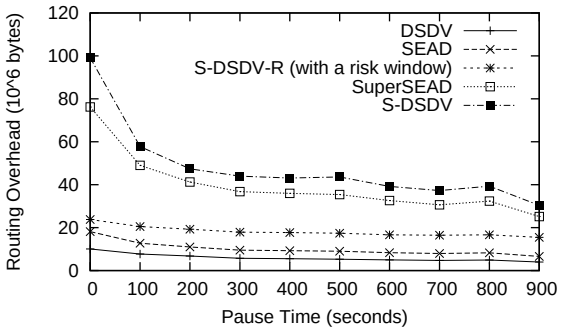

Fig. 2. S-DSDV-R offers better security than superSEAD with less network overhead, but bears a similar risk window of superSEAD.

8 Concluding Remarks

We propose the use of consistency checks for validating DSDV routing updates by additional messaging (i.e., by route requests and responses). In-band mechanisms (i.e., included within a routing update) are also possible, but would most likely involve generation and verification of digital signatures. Thus, it increases computational overhead

and may be subject to denial of service attacks. We note that similar ideas to those used in S-DSDV can be applied to secure other routing protocols.

Acknowledgments. The first author is supported in part by Alcatel Canada, MITACS (Mathematics of Information Technology and Complex Systems), and NCIT (National Capital Institute of Telecommunications). The second author is supported in part by MITACS and NSERC (Natural Sciences and Engineering Research Council of Canada). The third author is Canada Research Chair in Network and Software Security, and is supported in part by NCIT, an NSERC Discovery Grant, and the Canada Research Chairs Program.

References

[1] B. Awerbuch, D. Holmer, C. Nita-Rotaru, and H. Rubens. An On-Demand Secure Routing Protocol Resilient to Byzantine Failures. In *Proc. of WiSe'02*, September 2002.

[2] A. Barbir, S. Murphy, and Y. Yang. Generic Threats to Routing Protocols. Internet Draft (work in progress), April 13, 2004.

[3] S. Bellovin. Security Problems in the TCP/IP Protocol Suite. *ACM Computer Communications Review*, 19(2): 32-48, April 1989.

[4] K. Fall and K. Varadhan, editors. The *ns* Manual (formerly *ns* Notes and Documentation). April 14, 2002. http://www.isi.edu/nsnam/ns/doc/index.html

[5] J.J. Garcia-Luna-Aceves and S. Murthy. A Loop-Free Algorithm Based on Predecessor Information. In *Proceedings of IEEE INFOCOM'95*, Boston, MA, USA. April 1995.

[6] Y.C. Hu, A. Perrig, and D.B. Johnson. Ariadne: A Secure On-Demand Routing Protocol for Ad Hoc Networks. In *Proc. of MOBICOM'02*, September 2002.

[7] Y.C. Hu, D.B. Johnson, and A. Perrig. Secure Effi cient Distance Vector Routing Protocol in Mobile Wireless Ad Hoc Networks. In *Proc. of WMCSA'02*, June 2002.

[8] Y.C. Hu, D.B. Johnson, and A. Perrig. SEAD: Secure Effi cient Distance Vector Routing for Mobile Wireless Ad Hoc Networks. *Ad Hoc Networks Journal*, 1 (2003):175-192.

[9] M. Just, E. Kranakis, and T. Wan. Resisting Malicious Packet Dropping in Wireless Ad Hoc Networks. In *Proc. of ADHOCNOW'03*, October 2003. Springer Verlag, LNCS vol 2856, pp.151-163.

[10] G. Malkin. RIP Version 2. RFC 2453 (standard). November 1998.

[11] S. Marti, T.J. Giuli, K. Lai, and M. Baker. Mitigating Routing Misbehavior in Mobile Ad Hoc Networks. In *Proc. of MOBICOMM'00*, August 2000.

[12] A.J. Menezes, P.C. van Oorschot, and S. Vanstone. *Handbook of Applied Cryptography*. CRC Press, 1996.

[13] C.E. Perkins and P.Bhagwat. Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers. In *Proc. of the SIGCOMM'94*, August 1994.

[14] Y. Rekhter and T. Li. A Border Gateway Protocol 4 (BGP-4), RFC 1771, March 1995.

[15] R. Rivest. The MD5 Message-Digest Algorithm, RFC 1321, April 1992.

[16] B.R. Smith, S. Murphy, and J.J. Garcia-Luna-Aceves. Securing Distance-Vector Routing Protocols. In *Proc. of NDSS'97*, San Diego, USA. February 1997.

[17] L. Venkatraman and D.P. Agrawal. Strategies for Enhancing Routing Security in Protocols for Mobile Ad Hoc Networks. *J. of Parallel Distributed Comp*. 63(2): 214-227, Feb. 2003.

[18] T. Wan, E. Kranakis, P.C. van Oorschot. S-RIP: A Secure Distance Vector Routing Protocol. In *Proc. of ACNS'04*, June 2004. Springer Verlag, LNCS vol 3089, pp.103-119.

[19] M.G. Zapata and N. Asokan. Securing Ad Hoc Routing Protocols. In *Proceedings of the ACM Workshop on Wireless Security (WiSe 2002)*, September 2002.

[20] Y.G. Zhang, W. Lee and Y.A. Huang. Intrusion Detection in Wireless Ad-Hoc Networks. In *Proc. of MOBICOM'00*, August 2000.

[21] L. Zhou and Z.J. Haas. Securing Ad Hoc Networks. *IEEE Network Magazine*, 13(6), Nov/Dec 1999.