Title
An End-To-End Approach to Secure Routing in MANETs

Author
Garcia-Luna-Aceves, J.J.

Publication Date
2010

Peer reviewed

SECURITY AND COMMUNICATION NETWORKS
Security Comm. Networks. 2010; **3**:130–149
Published online 23 June 2009 in Wiley InterScience
(www.interscience.wiley.com) DOI: 10.1002/sec.121

An end-to-end approach to secure routing in MANETs

Stephen Dabideen[1*,†], Bradley R. Smith[1] and J. J. Garcia-Luna-Aceves[1,2]

[1]*Department of Computer Engineering University of California, 1156 High Street, Santa Cruz, CA 95064, U.S.A.*
[2]*Palo Alto Research Center (PARC), 3333 Coyote Hill Road, Palo Alto, CA 94304, U.S.A.*

Summary

Providing secure routing in mobile ad hoc networks (MANETs) is far more difficult than establishing secure routing in wired networks or static wireless networks. Node mobility and the relative scarcity of bandwidth render prior solutions ineffective. Solutions based on securing link or path information do not work well in MANETs because the dynamic nature of links requires extensive use of flooding to establish effective countermeasures. On the other hand, solutions based on hop-by-hop exchanges of distance information are easily compromised. Instead of trying to secure the ordering of nodes, we argue that secure routing in MANETs must be based on the end-to-end verification of physical-path characteristics aided by the exploitation of path diversity to increase the probability of finding secure paths. We apply this approach to the design of the Secure Routing through Diversity and Verification (SRDV) protocol, a secure routing protocol that we show to be as efficient as unsecured on-demand or proactive routing approaches in the absence of attacks. We prove that the countermeasures used in SRDV can defend against a variety of known attacks to routing protocols, including attacks involving collusion, and the fabrication and modification of routing packets. We also show the effectiveness of the end-to-end mechanisms *via* simulations. Copyright © 2009 John Wiley & Sons, Ltd.

KEY WORDS: security; ad-hoc networks; end-to-end

1. Introduction

Many security solutions have been proposed for routing protocols in mobile ad hoc networks (MANETs); however, to the best of our knowledge, a complete and efficient solution to secure routing in MANETs has not yet been attained. We argue that this is due to the interplay between signaling packets and data packets, as well as the dynamic nature of MANETs.

On-demand or proactive routing protocols based on the distributed computation of distances to destinations must disseminate signaling packets in which the routing metric to destinations is modified on a hop-by-hop basis, so that nodes order themselves with respect to destinations according to the routing metric (e.g., hop count). This empowers adversaries in a MANET to perform attacks by using false distance information to disrupt the ordering nodes try to establish for different destinations. This is especially problematic when nodes act in collusion with other nodes [1]. Because of the problems in securing distance-based routing protocols, most previous approaches to secure routing in MANETs have focused on securing entire paths from source to destination or have each node along the path secure the link it intends to use (e.g., References [2,3]). However this is not a viable approach for large

*Correspondence to: Stephen Dabideen, SOEGRAD 1156 High Street, Santa Cruz, CA 95064, U.S.A.
†E-mail: dabideen@soe.ucsc.edu

MANETs, because it leads to unsustainable flooding of control packets in a MANET. Section 2 presents a summary of the prior approaches for secure routing in MANETs, which reveals that, while the use of cryptography has been used to successfully counter many types of attacks, the solutions proposed to date do not guarantee that data packets are delivered, even if nodes comply with correct control signaling. In fact, many attacks are aimed at forcing data to be routed through adversary nodes, and once this is done they can perform denial of service, or disclosure attacks. It is possible that data can be routed, without any manipulation of the network, through adversaries and most previous work would provide no defense in this case. Such attacks can be best detected, and arguably can *only* be detected by end-to-end means. If these attacks were to occur when the known topology information is correct, then the best means of defense is path diversity.

We introduce the *Secure Routing through Diversity and Verification* (SRDV) protocol in Section 3. The goal of SRDV is to efficiently compute and use the shortest un-compromised paths available for the transmission of data through a network. SRDV accomplishes this by computing paths on-demand to minimize routing overhead, ensuring the correctness and freshness of signaling through the use of digital signatures, sequence numbers, and hash chain authentication, *verifying* the performance of these paths with end-to-end probing to detect compromised paths, and load-balancing over a *diverse* set of paths (the region of interest) to counter attacks once detected. SRDV accomplishes this while using comparable, if not less, overhead than many traditional unsecured approaches. Figure 1 summarizes the main components of SRDV.

Section 4 provides a security analysis of SRDV and shows that the countermeasures it employs (combination of end-to-end verification and path diversity with digital signatures and hash chains) ensures that independent or colluding attackers cannot manipulate or disrupt the computation and effective use of routes.

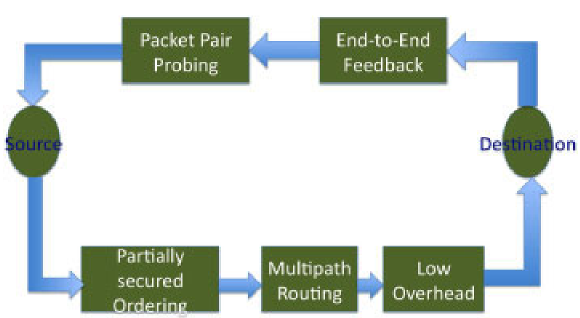

Fig. 1. SRDV components.

Table I. Notation.

Notation	Meaning
S	Source address of a data flow
D	Destination address of a data flow
RTT	Round trip time
s_X	Node X's secret (hash seed)
K^X	Node X's private key
SN_X	Current sequence number of node X
$\{X\}_K$	Formula X is encrypted under key K
$H_X(\cdot)$	Node X's hash function
$H^t_X(s)$	Secret s, hashed t times using $H_X(\cdot)$
$D_X(A)$	The distance (hop count) from X to A
t^o_{rtt}	Average RTT of RREQ-RREP pair
t^p_{rtt}	Average RTT of primary path (data)
t^s_{rtt}	Average RTT of secondary path (data)
l_p	Fraction of packets sent on primary path
l_s	Fraction of packets sent on secondary path

Section 5 presents the results of simulation experiments comparing the performance of SRDV with that of traditional nonsecure MANET routing protocols (Ad-hoc On demand Distance Vector Routing Protocol (AODV), Dynamic Source Routing Protocol (DSR), Optimized Link State Routing Protocol (OLSR)), as well as Authenticated Routing For Ad-hoc Networks (ARAN), which is a secure on-demand routing protocol based on distance information proposed recently. The results show that SRDV uses comparable, if not less, overhead than AODV, DSR, and OLSR in the absence of attacks. The simulation results also show that SRDV is capable of defending against a variety of attacks by independent and colluding adversaries.

The notation used throughout this paper is summarized in Table I.

2. Previous Work

Previous work on secure routing for MANETs has been based on mechanisms that either compromise scalability of the routing protocol, or leave routing vulnerable to significant attacks. Hu *et al*. [2] propose the *Secure Efficient Ad hoc Distance vector* protocol (SEAD) as an enhancement of the *Destination-Sequenced Distance-Vector* (DSDV) protocol [5] for secure routing in wireless networks. SEAD's primary enhancement over DSDV is the use of hash chains to authenticate the source of the update, and to secure the metric and sequence numbers contained in the update. There are a number of limitations with this solution. The use of a topology-driven routing model in which routes are pre-computed by all routers for all destinations in a network, is not a good match

DOI: 10.1002/sec

for mobile networks compared to on-demand routing protocols. In addition, the use of metrics updated by each hop in the network is susceptible to manipulation by compromised routers, and to colluding attackers (e.g., the 'Wormhole attack' [1]).

Hu *et al*. [6] also proposed the *Ariadne* protocol as an enhancement of the DSR protocol [7] for secure routing in wireless networks. Ariadne secures on-demand routing using a number of mechanisms that allow the source of the route request (RREQ) to verify that the request traversed a list of nodes given in the request, and that this list is the same seen by the target (destination) node when it received the RREQ. While Hu *et al*. present a security analysis of the protocol, a number of architectural limitations of the protocol remain. Using the end-to-end delay of signaling messages for route selection does not meet the requirements of network applications, which may have other performance constraints (e.g., bandwidth or reliability). More importantly, basing metric measurement solely on performance of signaling messages creates the vulnerability of attackers providing correct and expedited handling of signaling traffic while mis-handling data traffic. Lastly, the use of source routing inherited from DSR requires the potentially unnecessary communication of topology changes to all nodes using paths that include the affected topology, significantly limiting the efficiency of the protocol.

Eircksson *et al*. [3] propose the new *Secure Probabilistic Routing* (Sprout) protocol for secure routing in wireless networks, with the specific goal of protecting against colluding attackers. Sprout is a link-state protocol that uses probabilistic route generation and selection with end-to-end route performance feedback to secure the routing function. Probabilistic route generation and selection results in an inherently multi-path routing solution. The strength of Sprout is that it tends to find and use shorter routes exhibiting high delivery ratios over time, even in the context of compromised and colluding nodes. Its primary limitation is its dependency on global link state information or source routing.

Sanzgiri *et al*. [4] propose the *Authenticated Routing for Ad hoc Networks* (ARAN) protocol. ARAN is an on-demand routing protocol that uses hop-by-hop authentication of all routing messages (requests, replies, and errors) and end-to-end authentication of route discovery messages (requests and replies) combined with the use of an end-to-end metric to provide secure routing. The strengths of ARAN are that it is simple, ensures the authenticity and integrity of routing messages, and uses an un-spoofable end-to-end metric (delay of route signaling) to ensure loop freedom. However, ARAN has a few limitations. First, ARAN shares Ariadne's limitations on the use of a delay metric derived solely from signaling messages. Second, ARAN makes no provisions for the use of diverse paths in the event of problems with end-to-end data traffic performance, leaving it susceptible to attacks involving routers participating correctly in the routing computation but mishandling data traffic. Third, protocols that attempt to set up only a single path are particularly susceptible to attacks on route replies. If the route reply (RREP) is being unicasted from the destination to the source, as is the case of ARAN and AODV, it can be dropped without detection by an adversary along the path and hence future attempts at path discovery can lead to the same situation leading to a denial of service attack.

A number of solutions [8,9] have been proposed for securing routing in wired, non-mobile environments that derive from early work by Perlman [10]. In these solutions the routing computation is secured with the digital signature of link-state information by the node originating the link-state update. Receiving nodes validate updates before using them for their local computations. The limitations of this solution, discussed in Reference [9], is that compromised routers can advertise fabricated links, allowing arbitrary manipulation of the forwarding topology. One possible solution [9] is the use of authentication of each link by a routing authority for use in verifying the validity of a link advertisement. However, this solution is clearly not viable in a MANET environment.

3. SRDV

The goal of SRDV is to efficiently compute and use the shortest un-compromised paths available for the transmission of data through a network. SRDV accomplishes this by computing paths on-demand to minimize routing overhead, ensuring the correctness and freshness of signaling through the use of digital signatures, sequence numbers, and hash chain authentication, *verifying* the performance of these paths with end-to-end probing to detect compromised paths, and load-balancing over a *diverse* set of paths (the region of interest) to counter attacks once detected.

3.1. SRDV Overview

We present a brief overview of the protocol before diving into the specific details. SRDV is an on-demand

DOI: 10.1002/sec

routing protocol which uses RREQ and route replies to set up paths from the source to the destination. The route replies order a region of nodes with respect to the destination and proactive updates maintain the ordering in this region. Using packet pair probing techniques, bottleneck bandwidth is measured. Data packets are routed to the destination along multiple paths and the fraction of packets sent along each path is determined by the performance of each path.

3.2. Routing Establishment and Maintenance

The signaling in SRDV is hybrid, in that paths are established on demand and maintained proactively. When a node has data for a destination, it initiates a RREQ. This RREQ is flooded throughout the network and nodes derive and upon receiving a route request with a new sequence number, a node X records its current distance to the source $D_S(X)$ as well as the sequence number and its next hop in its routing table.

The destination responds to a route request by initiating a RREP. Route replies carry a new sequence number for the destination (the initiator of the RREP), the current distance traveled by the RREP and the distance from the source to the destination, $D_S(D)$ which is set by the destination itself and is based on the distance traveled by the RREQ. All nodes which receive a route reply packet will have an update value for distance to the source ($D_S(X)$), distance to the destination ($D_D(X)$) and the distance between the source and destination ($D_S(D)$). Using this information, a node will determine whether it is in the *region of interest* for this source destination pair as defined by the following equation:

$$D_S(X) + D_D(X) \leq D_S(D) + r \qquad (1)$$

where r is the number of retries since the last successful route discovery process.

Route reply packets are only forwarded by nodes which are inside this region of interest which is in contrast to protocols which sends the route reply to the source using unicast packets or those protocols which flood the route reply throughout the entire network. The result is the formation of multiple possible paths between the source and the destination while limiting the propagation of route replies to the region where they will be most useful.

The RREP is forwarded at most once by all nodes in this region. Like many routing protocols, a nonce (sequence number) is used to chronologically order control packets allowing nodes to distinguish new information from outdated information.

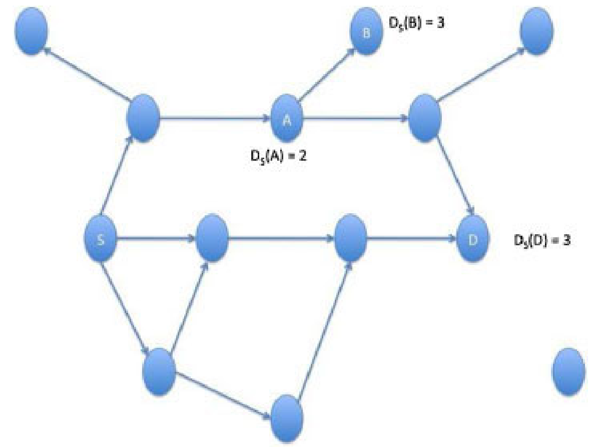

Fig. 2. Route request propagation.

As an example, consider Figures 2 and 3. They show the propagation of route requests and route replies in the network. To demonstrate how the region of interest is established, let us consider this to be the second attempt to establish a route after the first attempt had failed. This means that the value of r in Equation (1) is set to one (and this information is carried in the overhead packets). The route request is always propagated as shown in Figure 2 and all nodes record the distance to the source. The route reply is only propagated if the sum of the distance to the source and to the destination is less than or equal to four in this case. From Figure 3, it can be seen that Node A retransmits the route reply while Node B does not. It is clear that the nodes which forward the route reply would be a subset of nodes in the network.

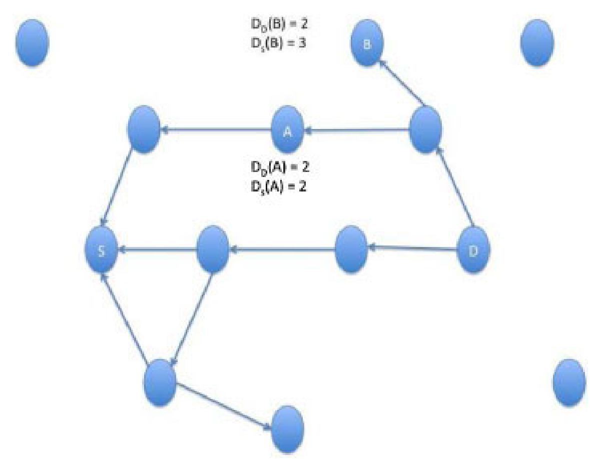

Fig. 3. Route reply propagaton.

DOI: 10.1002/sec

3.3. Maintaining the Region of Interest

The region of interest allows the possibility of a node having more than one successor to the destination and is necessary to achieve path diversity and allows for quick recovery in the event of link failures.

In a mobile environment, not only will links break but nodes will move into and out of the region of interest as defined by Equation (1).

Both the source and destination periodically initiate updates (unsolicited RREPs). The destination node sends periodic route replies every 10 s which serve to update the ordering of the nodes in the region of interest and the source sends these updates every 40 s as there is greater need to order the nodes with respect to the destination than to the source but the update from the source is needed to maintain the borders of the region of interest. These fixed values were used for simplicity, but a more dynamic update interval which depends on node mobility can be used.

These route reply messages will be propagated by nodes previously defined to be in the region of interest and the information will be stored, even if it is not forwarded, by those nodes along the border of the region of interest. When nodes move out of the region of interest they will no longer satisfy the constraints of Equation (1), and will no longer forward route requests and route replies. The periodic updates will ensure that all nodes within the region of interest, including those which recently moved into the region of interest, to maintain current distance information to both the source and destination which will allow then to become active and begin forwarding route request and route reply packets.

Algorithms 1 and 2 give the specific details on how RREQs and RREPs are handled respectively.

Instead of initiating a single RREP, a destination node occasionally initiates a pair of RREPs sent immediately one after the other. Each node in the region of interest uses this packet pair to estimate the bottleneck bandwidth to the destination through each of their neighbors from which it receives both packets. The precise manner by which this bandwidth estimation is achieved is presented later.

Nodes in a region of interest are ordered lexicographically according to their distances to the destination node, as well as the bottleneck bandwidth through each neighbor. This is attained by making nodes adhere to a *successor feasibility* condition, which mandates that successors in paths to destinations to have newer sequence numbers or the same sequence numbers but smaller hop counts to the destinations.

Algorithm 1 HandleRequest(RREQ)

if $AuthenticateSignature(RREQ) \neq TRUE$ **then**
  return
end if
S ← RREQ.destination.address
v ← RREQ.hashValue
H ← RREQ.hashFunction
Z ← RREQ.maxHashValue
C ← RREQ.hopCount
n ← RREQ.sourceSequenceNumber
if n ≤ latest source sequence number **then**
  return
end if
if $H^{d-C}(v) \neq$ Z **then**
  return
end if
UpdateRouteTable(S,n,C)
if RREQ.BlackList = 1 **then**
  temporarily blacklist primary successor
end if
if RREQ.BlackList = 2 **then**
  temporarily blacklist secondary successor
end if
if RREQ.BlackList = 3 **then**
  temporarily blacklist primary and secondary successors
end if
if $D_S(X) + D_D(X) \geq D_D(S) + 2$ **then**
  return
end if
RREQ.hopCount++
RREQ.hashValue = H(RREQ.hashValue)
BroadcastRREQ(RREQ)
return

3.4. Link Failures

When an intermediate node with a data packet experiences a link failure, it tries to route through a different neighbor that satisfies the successor feasibility condition. If there is no such a neighbor, the node changes its hop count to infinity and broadcasts an update to alert its neighbors of this change. Once neighboring nodes update their routing tables based on this information, they in turn check for an alternate feasible path. If a neighbor node finds a feasible path, it does nothing else. On the other hand, a neighbor with no feasible path changes its hop count to infinity and alerts its own neighbors with an update (a RREP). If the RREPs with infinite distances to the destination percolate to

DOI: 10.1002/sec

Algorithm 2 HandleReply(RREP)

```
if AuthenticateSignature(RREP) ≠ TRUE then
    return
end if
D ← RREP.destination.address
S ← RREP.source.adddress
v ← RREP.hashValue
H ← RREP.hashFunction
Z ← RREP.maxHashValue
C ← RREP.hopCount
n ← RREP.destination.SequenceNumber
P ← previousHop(RREP)
if n ≤ latest destination sequence number then
    return
end if
if H^(d−C)(v) ≠ Z then
    return
end if
UpdateRouteTable(D,n,C)
if isOdd(n) AND has received(n-1) then
    calculate BW to D through P
end if
if D_S(X) + D_D(X) ≥ D_D(S) + 2 then
    return
end if
if nodeId ≠ S then
    RREP.hopCount++
    RREP.hashValue = H(RREP.hashValue)
    BroadcastRREP(RREP)
end if
if nodeId = S then
    calculateRTTofOverhead(RREP)
    calculateRTTofData(RREP)
    HandleFeedback( )
    TransmitDataToD( )
end if
return
```

the source node, a new RREQ with a larger sequence number is initiated. Nodes that have reset their distances to a destination to infinity can change their hop count to finite values only upon receiving a RREP with a sequence number higher than the sequence number value for the respective destination they have stored in the routing table.

3.5. Digital Signatures

The distribution of certificates and keys are beyond the scope of this work and we assume the *a priori* distribution of keys to the nodes in the network through a trusted authentication server, much like in such previous work as ARAN [4].

Control packets are divided into two sections: the fixed fields and the incremental fields. The data in the fixed fields are set by the source of the packet, and while it may be read by intermediate nodes, it cannot be altered. The source uses its preassigned key to sign the fixed fields of control packets. This signature allows nodes to verify the source of the packet and to prevent nodes from tampering with the data.

The incremental fields of the control packets include the current hop count and a hash value, both of which are to be modified by each intermediate node. The exact fields in RREQs and RREPs are given in Table II.

3.6. Hash Chains

Each node, X, has a unique, cryptographically secure hash function $H_X(\cdot)$, such as MD5 or SHA-1, and generates a new random secret, s_X, each time it initiates a RREQ or RREP. It then calculates $H_i^d(s)$, where d is the maximum allowed length of a path. Care must be taken that none of the hash values based on this secret can be calculated based on past information. When necessary, a node can produce a new cryptographically secure hash function.

Control packets carry the current value, $H_X^t(s)$, where t is the number of hops from the source of the packet, along with the hash function itself and $H_X^d(s)$. Upon receiving a packet, the node checks the authenticity of the packet using the digital signature. Once authenticated, the node then verifies the integrity of the hash value by checking that

$$H^{d-t}(H_X^t(s)) = H_X^d(s) \quad (2)$$

where $H_X^d(s)$ is given in a fixed field, and therefore in the secure part of the packet, and $H_X^t(s)$ and t are

Table II. Fields of control packets.

	Fixed fields								Incremental fields	
RREQ	S	SN_S	D		$H_S(\cdot)$	$H_S^d(s)$	TTL	BlackList	$D_S(X)$	$H_S^{D_S(X)}(s)$
RREP	S	SN_S	D	SN_D	$H_D(\cdot)$	$H_D^d(s)$	TTL	Feedback	$D_D(X)$	$H_D^{D_D(X)}(s)$

DOI: 10.1002/sec

given in the unsecured part of the packet. If this equation does not hold, the packet cannot be trusted and is dropped without further processing. If the equation does hold, the routing table is updated and the packet is forwarded if it is in the region of interest. Whenever a node forwards a signaling packet, it hashes the current hash value $H_X^t(s)$ once using the given hash function and this new value will replace the old value before the packet is forwarded.

The simple consequence of this hash chain is that the hop count cannot be decreased without this attack being detected by the next node. Given $H_X^t(s)$ and $H_X^d(s)$ a node cannot calculate any value $H_X^x(s)$ for $x \leq t$ by virtue of the one-way hash function. This use of hash chains is similar to that in Reference [2], with the difference that sequence numbers are not covered by the hash chain, and the hash chain tail is included in each message.

3.7. Path Diversity

In a hostile environment, the probability of finding a secure path can be improved by using multiple paths. If one path performs noticeably worse than another, it could be due to adversaries on or near that path. Bad performance can be also attributed to benign causes, but neither case is desirable.

All the nodes in the region of interest are ordered with respect to their hop counts to the destination (the advertised metric) and bottleneck bandwidth (the measured characteristic) with the hop count having priority and bandwidth being used to choose between nodes with the same hop count.

In our implementation of SRDV we allowed nodes to use up to four paths and we compare the performance when fewer than four paths are used. Depending on the scenario, some node will have fewer than four successor. The source node assigns a label to each data packet which indicates to which successor the packet should be forwarded to (in terms the best successor, second best successor, etc.). Upon receiving a packet with label X, a node will forward it to its $(X \text{modulo} N)$th best successor where N is the current number of successors. This will not necessarily lead to node disjoint multi-path routing, because a node may only have one feasible successor, or because the best successor for one node may be the second-best for another. However, with a sufficiently populated region of interest, a node has multiple choices in successors and packets can travel along different paths. This is in contrast to many prior multi-path routing protocols which require the use of path information to ensure the paths are disjoint. This approach is also in contrast to node disjoint multi-path protocols such as MAODV.

The actual path which is defined by a specific label will vary over time. Measuring the performance of a specific path defined by knowing each node along the path is of little value in a mobile environment. The approach taken will allow the aggregation of many paths into a single label and the past performance of this group of paths can be more useful than the past performance of a very specific path.

3.8. QoS Probing

Nodes take measurements of delay and bandwidth to evaluate the validity of the advertised ordering (hop counts) with the actual performance of paths. Destination nodes immediately reply to RREQs by issuing RREPs. Hence, the time elapsed from the instant when a RREQ is initiated to the time the first RREP to it is received gives a good estimate of the end-to-end delay (t_{rtt}^o). The source node randomly selects data packets for which it will measure end-to-end delay. The end-to-end round trip time, t_{rtt} is given by

$$t_{\text{rtt}} = u_{\text{rcvd}} - d_{\text{sent}} - (u_{\text{sent}} - d_{\text{rcvd}}) \tag{3}$$

where:
d_{sent} = The time when data packet x was transmitted by the source.
d_{rcvd} = The time when x is received by the destination.
u_{sent} = The time when the first destination-based periodic update is issued after d_{rcvd}.
u_{rcvd} = The time that the update was first received by the source.

A simple comparison of the time experienced by the data packets and the RREQ/RREP can be used to detect attacks on the data packets. This process incurs no additional overhead because the extra information needed is carried in the periodic update packets.

The use of packet pairs to estimate the bottleneck bandwidth has been extensively studied [11–13] and more complex schemes have been since presented which use a packet train [14] or improved queuing [15]. Periodically, instead of initiating a single proactive update, the destination initiates two successive updates, with consecutive sequence numbers. Nodes in the region of interest measure the inter-arrival time of these packets and forward both packets. This inter-arrival time is taken to be inversely proportional to the bottleneck bandwidth. While this may be just a crude approximation, which may be called Asymptotic Dispersion Rate [16], it is sufficient for the purpose of

DOI: 10.1002/sec

comparing the expected performance of different nodes which are vying to be successors in a path to the destination. Successors with smaller inter-arrival times are preferred from a given a set of nodes with the same hop count.

3.9. End-to-End Feedback and Load Balancing

An important indication of performance is the number of packets delivered. In SRDV, packets travel along different paths and the destination records the number of data packets it received from each path based on the label of the data packet. An updated value for the number of packets received from each path is sent to the source in the periodic update packets or route reply message.

SRDV uses the feedback and path performance measurements to perform load balancing on the available paths.

The source of a data flow labels packets with one of two labels, which states the path that the packet should take. The performance of each path determines the fraction of packets sent along the path. The precise manner in which this feedback affects the routing of data packets at the source is shown in Algorithm 3.

Paths that deliver the greater fraction of packets are favored, as well as those with lower end-to-end data packet delivery time. If a node suspects that a given path is under attack based on measurements and feedback, it can set a blacklist flag in the next periodic update it issues. Upon receiving this notification a node ignores all overheard packets from its current successor depending on the flag (as seen in the algorithm). This allows for the formation of different paths the next time the ordering is updated.

Algorithm 3 HandleFeedback( )

x ← Primary Path Load Coefficient
y ← Secondary Path Load Coefficient
if $t^{o}_{rtt} \leq 2 * t^{p}_{rtt}\ OR\ l_p \leq 0.5$ **then**
    Set BlackList Flag for primary path
end if
if $t^{o}_{rtt} \leq 2 * t^{s}_{rtt}\ OR\ l_s \leq 0.5$ **then**
    Set BlackList Flag for secondary path
end if
if $t^{p}_{rtt} \geq t^{s}_{rtt}$ **then**
    $z \leftarrow 0.1 * l_s$
    $l_p \leftarrow l_p + z$
    $l_s \leftarrow l_s - z$
else
    $z \leftarrow 0.1 * l_p$
    $l_p \leftarrow l_p - z$
    $l_s \leftarrow l_s + z$
end if
if $l_p \geq l_s$ **then**
    $z \leftarrow 0.2 * l_s$
    $l_p \leftarrow l_p + z$
    $l_s \leftarrow l_s - z$
else
    $z \leftarrow 0.2 * l_p$
    $l_p \leftarrow l_p - z$
    $l_s \leftarrow l_s + z$
end if
return

4. Protocol Security Analysis

SRDV attempts to ensure that an attacker cannot manipulate or disrupt the routing computation. Manipulation of the routing computation allows an attacker to control the forwarding topology such that traffic is forwarded over paths containing the attacker. Given access to traffic, an attacker can launch denial of service, disclosure, or hijacking attacks on network sessions. Disruption of the routing computation results in various degrees of denial of service. The fundamental security requirements needed of a routing protocol to meet these goals are the authentication and authorization of nodes participating in the routing computation and the integrity and availability of the routing computation itself.

In the following we identify possible attacks on the routing protocol, characterize threats posed by these attacks, describe the countermeasures implemented in SRDV to eliminate or mitigate them, and prove that these countermeasures are sufficient for secure routing in MANETs.

In summary, authentication of route request and reply messages at each hop protects against the fabrication of these messages. Inclusion of a sequence number in route request, reply, and error messages protects against the replay of these messages. Authentication at each hop of route errors as coming from the next hop to the destination protects against the fabrication of route errors. Lastly, hash chains, and the end-to-end measurement of QoS parameters and delivery ratios protects against the fabrication of performance metrics and topology information.

We make the following three assumptions pertaining to the network:

DOI: 10.1002/sec

(1) The source and destination are not adversaries.
(2) The encryption process is secure and digital signatures cannot be forged.
(3) There is a path without any adversaries between the source to the destination at every instant. Otherwise, it would be impossible to secure the routing process.

4.1. Route Discovery

Route requests and replies can be deleted, fabricated, modified, or replayed by adversaries and this can lead to a variety of attacks.

Deleting a route request or reply prevents the discovery of an alternative path in the network. However, the path eliminated by this attack is a path that, by definition, contains an attacker. Furthermore, to have the ability to delete a routing message, the attacker must either be a compromised link or router. Protection from this class of attacker must come from external measures (link security, and system security of the router), and cannot come from the routing protocol itself. *Therefore, the best response is to allow this attack and avoid a known compromised path.* Protocols which set up only a single path, such as ARAN and AODV are more susceptible to this form of attack than SRDV since the overhead is used to set up multiple paths.

Fabricating a route request or reply results in resource consumption from the unauthorized flood of the request throughout either the network or region of interest, or the manipulation of the forwarding topology by an attacker masquerading as another source in the network. Manipulation can result in either denial of network service if the attacker drops packets for the advertised source, disclosure of traffic if the attacker chooses to forward traffic to the true source, or hijacking of sessions from a man-in-the-middle attack. Modification of the source and destination of a request results in a similar attack. *Authentication of the fixed fields in the request or reply at each hop are used as the countermeasure to this threat*. It is assumed that the encryption process is secure and digital signatures cannot be forged thus SRDV would be immune to this type of attack.

Replay of a route request or reply can result in the same compromises described above for fabrication. *The countermeasure to this threat is the use of a sequence numbers in route requests and replies*. Since the sequence number is in the fixed field, and therefore signed, it cannot to tampered with my intermediate nodes to make old packets appear new.

Lastly, *modification* of the hop count by an intermediate router results in the use of sub-optimal forwarding paths that include the attacker. This results in some unnecessary resource consumption, and the potential denial of service or disclosure of traffic described above. *Secure hash chains (Section 3.6) are used as the countermeasure to this threat*. This would be the most effective form of attack against SRDV, but we shall prove that adversaries are unable to prevent route discovery in SRDV regardless of their behavior.

Now we shall proceed to prove that the counter measures employed by SRDV can guarantee route discovery in the face of these potential attacks. We make the assumption that any node hashes the hash value in the route requests at least once before forwarding the request. If they were to forward the request without hashing it, at least one neighbor (the one it received the request from) would overhear the forwarded request with the same hash value and determine that the node is an adversary and would not route packets though it. If there is collusion, then this threat of retribution is reduced and such attacks could go undetected. Therefore, we also assume there is no collusion among nodes for this theorem to hold. While these assumptions may weaken the theorem, we later prove that without these assumptions SRDV will still guarantee the establishment of a path. We do take into account the possibility that malicious nodes may hash the value more than once, and this will go undetected. We use BAN Logic [17] for this proof. BAN Logic was designed to reason about authentication protocols and thus is useful for analyzing the countermeasures which guarantee authentication. In the following, the notation used is identical to that presented by Burrow's *et al*. [17] with the addition of the hash chain notation which is presented in Table I.

Theorem 4.1. *The use of Hash Chains in SRDV enables nodes to determine a lower bound on the length of the path from the source to the destination that is no smaller than the length of the path traveled by the route request.*

Proof. We make some simple extensions to the traditional BAN logic formalism to accommodate hash chains and multi-hop communications. Each node must hash the value received using the appropriate has function. We allow for the possibility of malicious node hashing it more than once; however, without collusion, a node cannot derive a previous hash value by nature of the hash function. Therefore, if a node knows how many times the value has been hashed it can deduce an upper bound on the length of the path (assuming no

DOI: 10.1002/sec

collusion) that is guaranteed to not be less than the true length of the path.

In our extension of BAN logic we introduce the *hash chain rule* that simply states

$$\frac{Y \text{ sees } H_X^i(s),\ Y \text{ believes } X \text{ believes } H_X^d(s)}{Y \text{ believes } i \geq D_X(Y)}$$

To translate the hash chain portion of SRDV into BAN logic we let Y_i range over nodes in a path with $i = 0 \ldots d$ (i.e., Y_i and Y_{i+1} are neighbors for $i = 0 \ldots (d-1)$), where Y_0 is the source of either a RREQ or RREP message, and let P be a nonce. The protocol now becomes:

$$\textbf{Message 1}: Y_i \rightarrow Y_{i+1} : \{P, H_{Y_0}^d(s)\}_K, H_{Y_0}^{i+1}$$

Additional fields in the overhead packet have been omitted since they bare no significance as to the validity of the hash chain authentication in the SRDV protocol. This message corresponds to forwarding of either a RREQ or RREP. The message includes the public key signature of a nonce (P) and the tail of the hash chain ($H_{Y_0}^d(s)$), and the hash computed by the previous node (Y_i) in the path. To analyze the protocol we first state the assumptions:

Assumption 1. Y_i believes Y_0 has public key K.

Assumption 2. Y_i believes Fresh(P).

The second assumption is somewhat surprising. It arises from the fact that, in the protocol, nonces are implemented as sequence numbers, and it is assumed each node remembers the last sequence number it saw in a request from a given node. Therefore, each node in the network is able to verify the freshness of the nonce in any routing message it receives.

Given these assumptions, the proof of the protocol proceeds as follows. After the transmission of Message 1 by Y_i to Y_{i+1} over one hop in the path, Y_{i+1} sees the message but does not understand it:

Step 1. Y_{i+1} sees $(\{P, H_{Y_0}^d(s)\}, H_{Y_0}^{i+1}(s))$.

Following the rules for decomposing a message, Y_{i+1} can now see the two components of the message:

Step 2. Y_{i+1} sees $\{P, H_{Y_0}^d(s)\}_K$.

Step 3. Y_{i+1} sees $H_{Y_0}^{i+1}(s)$.

Given Step 2, Assumption 1, and the *message-meaning* rule, Y_{i+1} can now deduce that Y_0 has signed the hash tail:

Step 4. Y_{i+1} believes Y_0 once said $(P, H_{Y_0}^d(s))$.

Note that Y_{i+1} does not yet know if the message is a replay or not. Given Step 4, Assumption 2, and the rule for generalizing freshness of part of a formula to freshness of the whole formula, Y_{i+1} can now deduce that the hash chain tail is fresh:

Step 5. Y_{i+1} believes Fresh$(P,\ H_{Y_0}^d(s))$.

Given Steps 4 and 5 and the *nonce-verification* rule, Y_{i+1} can now deduce that Y_0 believes the statement composed of the nonce and the hash chain tail (in the sense that Y_0 uttered the pair during the current epoch):

Step 6. Y_{i+1} believes Y_0 believes $(P,\ H_{Y_0}^d(s))$.

Given Step 6 and the rules for decomposing belief of compound statements, Y_{i+1} can now deduce that Y_0 believes (in the sense described above) the hash chain tail:

Step 7. Y_{i+1} believes Y_0 believes $H_{Y_0}^d(s)$.

Finally, Steps 3 and 7, with the *hash chain rule* presented above can be used to deduce that the path traversed by the request has at most $i+1$ hops, and that $i+1$ is no less than the true length of the path:

Step 8. Y_{i+1} believes $i + 1 \geq D_X(Y_{i+1})$.

Therefore, at each hop along a path each node Y_i is able to verify that the hash chain tail did in-fact originate at the source, Y_0, of the message. Furthermore, using the hash chain rule, Y_i can determine an upper bound on the hop count for the path traversed by the request that is guaranteed (assuming no collusion) to not be less than the true length of the path (in the event there are no adversaries, i from the request will be the true hop count of the path). ■

The length of the path is important in SRDV because after a certain point, routing packets flooded in the region of interest by one end of the data flow would be guaranteed to arrive at the other end, and this is integral to the end-to-end mechanism in SRDV.

Theorem 4.2. *Let $L_S^{t^n}$ denote the length of the shortest path $(N_1, N_2 \ldots N_k)$ between N_1 and N_k such that each N_i is not an adversary, at a time n. Let $L_R^{t^n}$ denote*

DOI: 10.1002/sec

the diameter of the region of interest between N_1 *and* N_k. *Then* $L_S^{t^n} \leq L_R^{t^n}$ *is a sufficient, but not necessary, condition to ensure that packets flooded in the region of interest by* N_1 *will be received by* N_k *and vice versa.*

Proof. The proof is by induction on the length of the path $(N_1, N_2 \ldots N_k)$.

For a path of length 1, N_1 transmitted the packet, and N_2 is a neighbor of N_1, therefore N_2 would have received the packet (from N_1), so it is true for a path length of 1.

Now assume it is true for a path of length j, where $0 \leq j \leq L_R^{t^n}$. Since $j = L_S^{t^n} \leq L_R^{t^n}$, N_j must be in the region of interest and is not an adversary so N_j would retransmit the packet. Therefore the packet would be received at N_{j+1}.

The same argument can be used to prove the reverse direction, any packet flooded in the region of interest of N_k would be received by N_1.

The condition is not necessary, because packets can arrive at N_k from N_1 through a possibly shorter path that contains adversaries. ■

Theorem 4.3. *Adversaries cannot indefinitely prevent route discovery in SRDV.*

Proof. To prevent route discovery between source N_1 and destination N_k, node N_1 cannot receive a RREP for a RREQ it issued. There are two possible cases:

(1) The RREQ never arrived at the destination.
(2) The RREQ arrived at the destination D, but the RREP never arrived at the source of the RREQ.

For the first case, the diameter of the region of interest is the diameter of the network therefore $L_S^{t^n} \leq L_R^{t^n} = \textit{Network Diameter}$, and by Theorem 4.2, the RREQ would arrive at the destination. Therefore, this case is not possible.

Consider the second case. If N_1 did not receive the RREP, it will retry the RREQ and we can be certain this RREQ will reach N_k. At this point, N_k would set the diameter of the region of interest to $L_R^{t^{n+1}} = L_{\mathrm{RREQ}}^{t^{n+1}} + r * k$ where r is the number of retries, $k > 0$, and $L_{\mathrm{RREQ}}^{t^{n+1}}$ is the distance traveled by the route request, at time $(n+1)$. Since $L_{\mathrm{RREQ}}^{t^{n+1}} \geq 0$, we have $L_R^{t^{n+1}} \geq r * k$.

Assume for contradiction that, for all values of r, that $L_S^{t^{n+r}} > L_R^{t^{n+r}}$. However, $L_R^{t^{n+r}} > r * k$, which is not bounded. Let $r = D$ equal the network diameter. Substitution gives, $L_R^{t^{n+r}} > D * k > D \geq L_S^{t^{n+r}}$ which gives the desired contradiction.

We note that eventually, $L_S^{t^{n+r}} \geq L_R^{t^{n+r}}$ after some number of retries, and at this point, by Theorem 4.2, we can be assured that the RREP will arrive at N_1 at which point in time route discovery would have taken place. ■

While the hash chain does not completely prevent nodes from advertising false hop counts, it makes it more difficult to do so. A node cannot reduce the hop count advertised to a value lower than that it received in a request or a reply, but it could potentially forward the packet without hashing the value of after hashing the value more than once. With each retry, more nodes along the path must perform this attack to prevent the RREP from arriving at the source and there is a limit to which this can be done. For a denial of service attack, the adversaries must ensure the conditions in Theorem 4.2 are never satisfied but Theorem 4.3 shows that with iteration there must be more nodes in collusion to prevent discovery and therefore shows the existence of a limit to this attack.

Protocols that attempt to establish a single path, as is the case with ARAN and AODV, an adversary along the path could drop RREP packets and this would lead to an uncorrectable denial or service attack.

4.2. Route Maintenance

Route error messages (RERR) can be deleted, fabricated, modified, or replayed. *Fabrication* or *modification* of a route error by an attacker not on the current forwarding path results in the redirection of traffic to a sub-optimal path, and possibly to a path containing the attacker. This can result in unnecessary resource consumption, denial of service, or disclosure of data traffic. *Authentication at each hop of the RREQ as having come from the next hop neighbor to the destination protects against this attack.*

Replay of a RERR has the same effect as fabrication, and is protected against by the *inclusion of a sequence number in route errors.*

Deleting a RERR results in resource consumption and denial-of-service from the transmission of packets along a dead-end path. *The end-to-end feedback and load-balancing mechanisms described in Section 3.9 are used as countermeasures to this attack.* Protocols without feedback can potentially send all their packets to an adversary after a link failure without knowing.

Theorem 4.4. *Attacks on RERRs cannot permanently disrupt routing in SRDV.*

DOI: 10.1002/sec

Proof. A node cannot successfully fabricate of modify a RERR from another node, given that we assume the encryption process is secure; therefore, adversaries cannot produce the digital signature necessary to fabricate RERR packets or resign modified RERR packets. Using sequence numbers, which cannot be modified because they are encrypted, prevents the successful replay of old RERR packets. This leaves only RERR fabrication or deletion as the two possible attacks. An adversary can fabricate a RERR for a link to one of its neighbors, but multiple paths are used and this would eliminate a path with an adversary and will thus not disrupt routing. An adversary can drop RERR. This can potentially lead to data packets being routed to a node with no path to the destination, and that node will therefore drop the data. By Corollary 1 we know feedback information will eventually reach the source of the data flow and once detected, the situation would be corrected by load balancing or finding new paths. Therefore, no possible attacks on RERRs can permanently disrupt routing in SRDV. ■

4.3. Securing Data Delivery

Securing route discovery and route maintenance is essential to successfully routing data, but by itself would prove to be an insufficient solution. The routing protocol should be able to detect and avoid malicious attacks on data packets. Some nodes may behave correctly during the route discovery phase but then drop data packets routed through them, or they may use a wormhole, which is undetectable in the route discovery phase to force packets to be routed through them and then perform denial of service or disclosure attacks. The most reliable means to detect such attacks on data packets is to through end-to-end feedback. Corollary 1 proves that the performance feedback reaches the source node, and this is crucial to detecting attacks. Adversaries may be able to temporarily disrupt the feedback mechanism, but this action cannot be maintained indefinitely.

Corollary 1. *Feedback information from the destination eventually arrives at the source.*

Proof. A destination node N_k can determine if its update packets (with feedback information) arrive at the destination based on the sequence number for N_k in the update from the source N_1 or the lack of such an update. Once the destination determines the updates are not being received at the source, it can increase the diameter of the region of interest until update packets are delivered and from Theorem 4.3 it follows that this must happen. ■

The assurance that feedback is received by the source then ensures that attacks on data packets must be detected. Once an attack on the data is detected by the source, there are two possible actions.

(1) If the attack involves only one path, the source node can send a greater fraction of packets through another path.
(2) If the attack involves all the paths being used, then the source can set the 'blacklist flag' in a RREQ, and this forces nodes in the network to choose different successors, which results in a potentially different ordering of the nodes in the network.

The number of times the source forces nodes to blacklist their neighbors is limited to two times in our implementation, because doing this too many times can partition the network, making it impossible to deliver packets. The use of the blacklist flag attempts to create different paths that do not involve an adversary that is currently attacking data packets. However, the success of this mechanism cannot be guaranteed and this issue is a matter for future work.

End-to-end coordination is necessary to detect and correct malicious behavior and the design of SRDV ensures that paths from the source to the destination are established and their performance are monitored (in terms of packet delivery and delay). It ensures that these measurements are delivered to the source node, which can then adjust its behavior in an attempt to improve performance.

4.4. The Security of SRDV

We now prove that the various security mechanisms of SRDV will together ensure the security of SRDV. Specifically, countermeasures include the digital signature of fixed fields in signaling messages (Section 3.5), protection of hop count using hash chains (Section 3.6), measurement of round-trip-time and bottleneck bandwidth using QoS probes (Section 3.8), and the use of diverse paths based on delivery ratios (Sections 3.7 and 3.9).

Theorem 4.5. *The countermeasures employed by SRDV results in secure routing.*

Proof. This proof is by contradiction. Assume the countermeasures are functional and effective, and that, nonetheless, security has been compromised.

DOI: 10.1002/sec

Compromised security is interpreted in the most general terms as control of the forwarding path by an attacker using falsified information, or the use of a compromised path to disrupt communication. Focusing first on the manipulation of the path selection process, three metrics are used to select paths in SRDV: hop count, QoS measures of round-trip-time and bottleneck bandwidth, and packet delivery ratio. Two possible approaches for manipulating these metrics include direct modification, or masquerading as a traffic source. Both approaches support building forwarding paths to a destination that include an attacker based on false path information.

Assuming the implementation of the countermeasures as described above, there are limited opportunities for manipulating the metrics used for path selection. As described in Theorem 4.1, an attacker can only increase hop count. This would allow an attacker on the path resulting from an uncompromised routing computation to repel traffic to another path. However, such an attack is pointless in the sense that it requires a compromised path to launch an attack (i.e., an attacker happens, by luck, to be on the path of desired traffic). The analysis of such attacks, based on compromised paths, is presented below. Similarly, the round-trip-time and bottleneck bandwidth metrics computed by the QoS probes can only be made to look worse by an attacker, with the same assessment given above for increasing hop count. Therefore, compromise of the routing computation in any meaningful way (i.e., that does not require a pre-existing compromised path) from the direct manipulation of metrics requires failure of one of the hash chain, QoS probes, or diverse path countermeasures, which contradicts the assumptions. Even amidst colluding nodes, there a limit to the manipulation (Theorem 4.2) and path discovery is guaranteed (Theorem 4.3). Furthermore, once a path is established, adversaries cannot manipulate error messages to disrupt routing (Theorem 4.4).

Manipulation of path selection metrics by masquerading as the traffic source requires fabrication of the source address of a signalling message. These messages are protected by a digital signature that is assumed to be active and effective. Therefore, compromise of the routing computation by masquerading as the source requires failure of the digital signature countermeasure, which contradicts the assumptions.

Assuming the existence of a compromised path, communication can be disrupted either from the dropping or delay of traffic (e.g., delay resulting from processing required for a man-in-the-middle attack). These attacks would be detected (Corollary 1) by either the delivery ratio or QoS probing countermeasures. Therefore, disruption of communication over a compromised path requires failure of the countermeasures, which, again, contradicts the assumptions.

In summary, for SRDV to be compromised, one of the countermeasures must be disabled or compromised, which contradicts the assumptions. ■

Note that this proof is robust in the presence of compromised routers with collusion, except for the attack where a compromised node masquerades as a source router. In this case protection from the attack will depend on detectable disruption of traffic flow from the use of a non-optimal path.

5. Simulations

We use simulation experiments to show that, in the absence of attacks, SRDV can be as effective as proactive and reactive routing protocols with an insignificant increase to the routing overhead required. The results from the experiments also illustrate that SRDV is capable of defending against a variety of attacks in hostile environments.

We compare the performance of SRDV to that of AODV, DSR, OLSR and ARAN. While we acknowledge that there are several protocols which have been shown to outperform these, AODV, DSR and OLSR are the most popular representatives of on-demand and proactive routing, and ARAN is a good example of secure routing protocols based on distance vectors. Another type of routing protocol is hierarchical routing protocols such as ZRP [18]. Such protocols consists of proactive regions or domains joined together by an inter-domain routing protocol. The security of such protocols would be effectively as good as the security of the intra-domain routing protocol which is usually proactive. Therefore comparisons to proactive protocols should suffice in a thorough analysis.

In addition to simulating SRDV, we simulate uSRDV, the unsecured version of SRDV. In uSRDV we remove the multi-path capabilities, the end-to-end feedback and measurements, the cryptography and the hash chains. This leaves a basic, single-path hybrid routing protocol. Using this as a base measure, we can highlight the cost of our security mechanisms. Throughout the paper we argue the need for path diversity. To support this argument we use two variations of SRDV: one which uses two paths (called SRDV2) and one which uses four paths (called SRDV4).

DOI: 10.1002/sec

Table III. Simulation parameters.

Parameter	Value
Simulation time	900 s
Number of nodes	100
Simulation area	1000 m × 1000 m
Node placement	Uniform
Mobility model	Random waypoint
Min–max speed	1–10 m/s
Pause time	30 s
Propagation model	Two-ray
Physical layer	802.11
Antenna model	Omnidirectional
MAC protocol	802.11 DCF
Data source	CBR
Number of packets per flow	800
Packet rate	4 packets per second
Node density	0.001 nodes/m^2
0	0

All the experiments were performed in Qualnet 4.5. The versions of AODV, DSR and OLSR were the versions built into Qualnet. We implemented ARAN based on the details provided in Reference [4]. Three scenarios were used and the parameters are summarized in Table III and described in the following subsections.

Each experiment lasted 900 s and for each protocol the experiment was repeated 50 times with random node placement and mobility. There were 10 CBR sources in Scenarios A and B and 20 CBR sources in Scenario C, which started generating packets at a random time to a randomly chosen destination. Each CBR source generated 800 packets at a rate of 4 packets per second.

5.1. Scenario A

Scenario A was designed to test the performance of the protocols in a dynamic environment with volatile links. This choice of parameters satisfies the minimum standards for rigorous MANET protocol evaluation as prescribed in Reference [19], because it results in an *average shortest path hop count* [19] of 4.03 and *average network partitioning* [19] of 3.9%. This ensures that packets travel several hops from source to the destination and thus tests the robustness of the protocols. 100 nodes were randomly distributed over an area of 1000 m × 1000 m with the radio range set to 150 m.

5.2. Scenario B

The second scenario uses the same area and number of nodes as Scenario A but the node have greater radio range of 200 m. This increases the number of neighbors and extends the lifetime of paths as nodes take a longer period of time to move our of range of each other. Consequently, the average network partitioning as well as the average shortest path hop are reduced. This scenario is interesting because it increases the number of paths between the source and destination and the SRDV routing protocol was designed to take advantage of this path diversity.

5.3. Scenario C

The third scenario was designed to test the performance of the routing protocol in a larger network with more flows and a varying number of adversaries. The network consists of 200 nodes uniformly distributed over a 1000 m × 2000 m region with 20 CBR flows initiated at a random time between a randomly selected source and destination pair. This will increase the load in the network and the expected length between the source and destination pairs. We vary the number of adversaries and observe the performance of the protocols.

5.4. Evaluation Metrics

Three metrics were used to evaluate and compare the performance of the protocols. Delivery ratio is the fraction of packets that arrive at the corresponding destination by the end of the simulation. Latency is the average end-to-end delay experienced by the data packets. Net load is the number of control packets (RREQs, RREPs, RERRs, Hellos, and TC messages) which were initiated or forwarded, divided by the number of data packets sent. This last metric gives an indication of the average number of control packets needed to send a packet from the source to the destination.

5.5. Performance with No Adversaries

The first set of experiments aims to show the effectiveness of the SRDV protocol in an environment where there are no attackers. Each node uses the correct information in the update packets and tries to deliver every packet to the destination. The simulation results for the five routing protocols tested are summarized in Table IV, where the mean and a 95% confidence interval are given.

The introduction of security mechanisms usually have detrimental impact on the performance of routing protocols. The mechanisms increase the overhead and complexity of the protocol and care should be taken

DOI: 10.1002/sec

Table IV. Simulation results: no attacks.

	Scenario A			Scenario B		
	Delivery ratio	Latency	Net load	Delivery ratio	Latency	Net load
AODV	0.60 ± 0.10	0.09 ± 0.04	14.4 ± 5.3	0.90 ± 0.03	0.072 ± 0.015	5.04 ± 1.31
DSR	0.14 ± 0.10	18.5 ± 15.9	5.0 ± 1.2	0.14 ± 0.04	1.3 ± 0.87	5.9 ± 0.76
OLSR	0.30 ± 0.08	0.07 ± 0.03	67.5 ± 1.2	0.71 ± 0.04	0.104 ± 0.021	17.2 ± 0.2
ARAN	0.53 ± 0.09	0.21 ± 0.11	24.7 ± 5.0	0.91 ± 0.07	0.11 ± 0.06	3.0 ± 0.9
uSRDV	0.78 ± 0.10	0.15 ± 0.10	7.9 ± 2.7	0.98 ± 0.03	0.067 ± 0.047	1.92 ± 0.20
SRDV2	0.69 ± 0.08	0.10 ± 0.04	3.9 ± 0.6	0.92 ± 0.02	0.05 ± 0.02	3.9 ± 0.4
SRDV4	0.74 ± 0.07	0.11 ± 0.05	3.9 ± 0.6	0.89 ± 0.03	0.07 ± 0.018	3.42 ± 0.19

that the addition of security does not render the protocol ineffective because the ultimate goal is to deliver packets to the destination.

These experiments demonstrate that the addition of the security mechanisms in SRDV has minimal impact on its efficiency. By comparing the performance of uSRDV to SRDV we do notice that less packets are delivered and the overhead of SRDV is almost twice as much as uSRDV. However, when comparing the performance of SRDV to other protocols such as AODV, OLSR, DSR and ARAN we see that despite the degraded performance relative to SRDV, it still outperforms these other protocols.

5.5.1. Overhead

SRDV uses packet pair techniques to measure physical path characteristics and this results in more route reply messages especially since this is done on a periodic basis. Furthermore, when a source node set the blacklist flag, it initiates a route request flood even if there is a path which also contributes to the additional overhead. Despite all of this, SRDV maintains comparable and in some cases significantly lower overhead than other protocols. Since SRDV sets up multiple paths between the source and destination in the event of link failures another path my be already available and this means that there is no need to flood the network. In contrast ARAN and AODV sets up single paths and in a mobile environment, this will eventually fail and these protocols have no option but to flood the network again. OLSR, being a proactive routing protocol, uses a lot of hello messages and topology control packets and adds significant overhead.

5.5.2. Delivery ratio

All protocols perform better in Scenario B than in Scenario A and this is especially apparent in terms of delivery ratio. The increased radio range (from 150 m in Scenario A to 200 m in Scenario B) resulted in less link failures and more packets being delivered. SRDV uses periodic proactive updates to re-order the nodes in the region of interest between the source and the destination. This up-to-date ordering would be more effective than the ordering which is only establish once when the source floods route requests as in AODV and ARAN.

5.5.3. End-to-end delay

The establishment and use of multiple paths in SRDV has can have both positive and negative impacts on the end-to-end delay. Unlike AODV and ARAN which drops route packets and initiate a new route request upon link failures, SRDV attempts to route though the known ordering among nodes in the region of interest. This ordering is based on the last route reply (or proactive update). If the links are still valid, then SRDV will deliver packets with lower end-to-end delay than protocols which use the route discovery process immediately after link failure. However, if the ordering is no longer valid, it may take longer for all the invalid links to be discovered before the route discovery process is initiated and this results in greater end-to-end delay.

ARAN is, to some extent, a simplified version of AODV in which many of the optimizations have been removed since they introduce security risks. Without attacks, its performance is worse than AODV with respect to packets delivered and delay in scenario A but comparable in scenario B. But this highlights the inefficiency of the protocol when there are no attacks, and this is primarily because only a single path is set up and only the destination node can reply to RREQs.

5.6. Independent Adversaries and Varying Mobility

In this set of simulations, we use Scenario B and allow for 30% of the nodes to be attackers on average, but each acts independently of the others. We vary the

Table V. Simulated Node Behavior.

Percentage	Behavior
70	Well-behaved (best effort routing)
10	Drops RREPs
5	Increases hop count, drops data
5	Increases hop count, forwards data
5	Decreases hop count, drops data
5	Decreases hop count, forwards data

mobility of the node and observe the performance of the protocols. These attackers modify the hop count by a random amount, drop data packets and drop control packets (RREPs in particular). This results in a wide variety of attacks with the goal of either capturing data packets or preventing data packets from reaching the destination (not by capturing the packets but by thwarting the routing process). There is no merit in simulating fabrication and masquerading attacks, because the digital signatures render these attacks futile. The breakdown of percentage of specific types of attackers is given in Table V. We first give an intuition of the effects of these types of attacks and then support this with simulation results.

For comparison, we use an authenticated form of AODV (which we call aAODV), which requires nodes to sign packets they initiate. This protects attacks in which nodes fabricate RREPs and/or masquerade as other nodes. For this to work, intermediate nodes cannot issue RREPs in response to RREQs, given that malicious nodes could respond to all RREQs and force data to flow to those nodes, even if the malicious nodes do not have valid paths. The addition of digital signatures and the disabling of intermediate replies are the only differences between AODV and aAODV. We must emphasize that aAODV is not a novel approach and similar proposals have been made in the past. The relevance of using aAODV in our comparisons is twofold. First, using AODV is not adequate, because it simply becomes inoperable under attacks, which is also the case for DSR and OLSR. Second, and more importantly, aAODV and SRDV utilize these same authentication services; therefore, the difference in performance between the two protocols can be attributed to the path diversity and the end-to-end feedback mechanisms that we want to highlight.

5.6.1. Analysis of these attacks on routing protocols

In AODV, and therefore aAODV, the RREP is returned to the source along a single path which was set up by the RREQ and is therefore independent of hop count. Consequently, aAODV is immune to attacks that attempt to modify the hop count. This is not the case with uSRDV, which attempts to set up multiple paths and therefore relies on hop counts to order the nodes into successors and predecessors. By modifying the hop count in uSRDV, the ordering of nodes will not reflect the actual topology, and can lead to loops or packets being routed to dead ends. On the other hand, given that RREPs travel along single paths in AODV and ARAN, these and similar protocols are particularly vulnerable to attacks over the single paths traversed by RREPs. If an attacker lies on this 'best' path and it forwards RREQs but drops RREPs, the consequence is repetitive flooding of RREQs by the sources, which quickly degrades the performance of the network by consuming excessive amounts of bandwidth with RREQs. This is not the case in SRDV! If an attacker refuses to forward a RREP, chances are that its neighbors will and hence the data packets will not be routed through the attacker.

The aAODV protocol has no protection against malicious nodes that forward control packets but drop data packets. Given sufficient multi-path options, SRDV sends the greater number of data packets along the more successful routes. However, the ordering in SRDV can be compromised, which could be another reason why packets do not arrive at the desired destination.

DSR requires overhead packets carry the route traveled which makes it difficult to modify the hop count without the attack being detected. However, without authentication, any node can pretend to be adjacent to the destination or even pretend to be the destination and this cannot be detected. Furthermore, DSR, like AODV and ARAN has no protection against nodes which forward overhead packets but drops data packets.

OLSR can be particularly vulnerable to nodes which modify the hop count to destinations in its proactive updates. Such false information can be disseminated throughout the network even before the destination is needed and this can severely degrade the performance of the network. Attackers which drop overhead packets would have less impact on OLSR than the other protocols because in a well connected network the updates would arrive at any given node though several other nodes.

5.6.2. Delivery ratio

The delivery ratio of the protocols with varying pause times are shown in Figure 4. It is an interesting result that with all the protocols tested, as the mobility of

DOI: 10.1002/sec

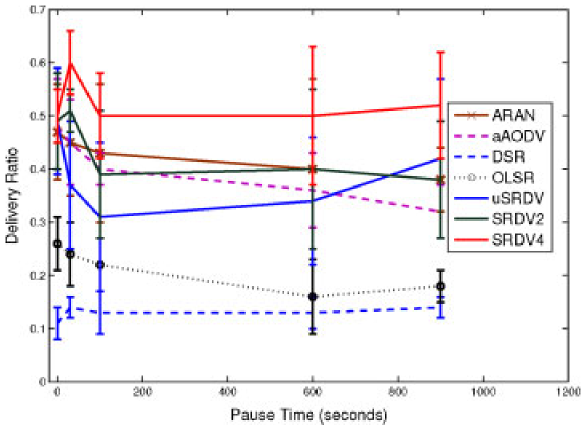

Fig. 4. Delivery versus pause time.

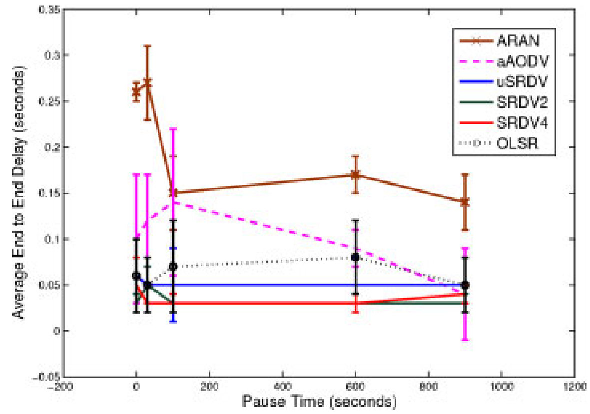

Fig. 5. Delay versus pause time.

the nodes decreases the number of packets delivered also decreases. As the nodes move about, paths will constantly be changing and sometimes there will be an attacker on the path capable of disrupting the specific routing protocol. Most of the protocols tested have no means of detecting attacks and if the network is almost static and there is a node along the path which is dropping data packets then there is no way for the protocols (such as ARAN, AODV, DSR and OLSR) to correct this. SRDV on the other hand uses end-to-end feedback and will detect such attacks. Packets will be lost before it is detected, but it has the potential to blacklist such adversaries before waiting for a link failure. An opposing factor to this is that as mobility increases, the efficiency of the routing protocol decreases and this explains the shape of the curves (with a minimum point in the middle) . It becomes evident that in a wide variety of mobility scenarios SRDV out performs the other protocols and as number of paths used by SRDV increases, the performance also increases.

5.6.3. End-to-end delay

For all the protocols tested, there was very little variation in the average end-to-end delay for the different pause times as seen in Figure 5. It is clear that ARAN's average end-to-end delay is larger than that of AODV which is much larger than that of SRDV. AODV and ARAN are particularly susceptible to attackers which drop route reply messages since these messages are sent on a single path. When this happens, the source of the flow may have to initiate the route discovery processes several times before a route without such attackers is discovered which causes the comparably larger delays. In SRDV however, the route reply messages travel along many different paths from the destination node to the source node and unless there is an attacker which drop route reply messages along each of these paths it would be impossible to deliberately prevent route discovery. The delay associated with DSR was far greater than the others was was beyond the borders of the graph. OLSR also experienced very small delays, but the packets that were actually delivered were those where the source and destination were two hops or less away from each other.

5.6.4. Overhead

The results for the overhead incurred by the protocols is shown in Figure 6. In this scenario, the total number of control packets depends on the types of attackers on the path. Adversaries which forward overhead but drop data packets can decrease the number of control packets used. The path beyond such an adversary could break but the source will never learn of this. An extreme example is where the adversary is one hop from the

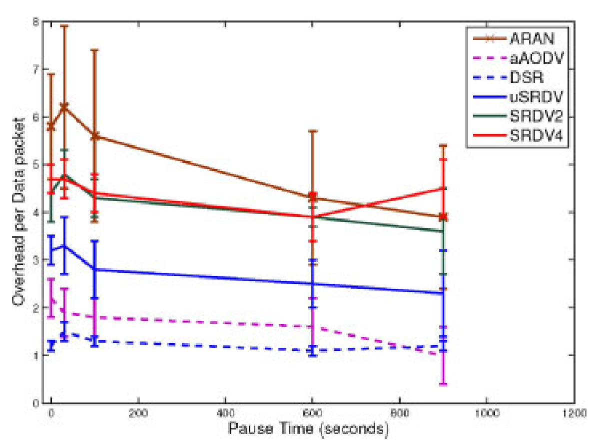

Fig. 6. Overhead versus pause time.

DOI: 10.1002/sec

source and drops all data packets. It is possible that the source can send all its data packets to the node which then get dropped as long as they remain neighbors and the attack is not detected. On the other hand, adversaries which drop overhead packets (especially route replies) can prevent the route discovery process which can result in repeated flooding of the network which will greatly increase the routing overhead experienced by the network. The general trend is that as the pause time increases the overhead decreases. This is usually the case as paths break less often and there is less need for route repair. The use of packet pair probing and proactive updates in SRDV results in greater overhead than aAODV but still not as much as ARAN.

5.7. Performance with Increasing Number of Adversaries

The objective of this scenario is to observe the protocols as the number of adversaries increase in the network. We use Scenario C: a 200 node network, with 20 flows starting at a random time and sending 800 packets from a randomly chosen source to a randomly chosen destination. We vary the number of Adversaries from 0 nodes to having 90 out of the 200 nodes perform some type of attacks. Of the adversary nodes, one-third drops data packets, one-third drops overhead packets and the remaining one-third alter the hop count in overhead packets. It is expected that as the number of adversaries increase, the performance of all the protocols will decrease. In a network where almost every other node is an attacker of some form, routing will indeed be difficult.

5.7.1. Delivery ratio

One limitation to SRDV is its simplicity. Groups of paths are weighted together as nodes move about and an entire path is blacklisted. The ideal situation would be do discover the precise attacker and blacklist that node or choose a route where all the nodes are known to perform by the source. However, this would require complete path information and even source routing and this is a not a viable option in a mobile environment. SRDV can set the blacklist flag up to two times. The continual use of this option can result in network partition and in this case even fewer packets will be delivered. As the number of adversaries increase, the likelihood of an adversary being on the path increases and SRDV can detect and attempt to correct this up to two times. While this is sufficient for a smaller number of adversaries, the gap in performance does decrease

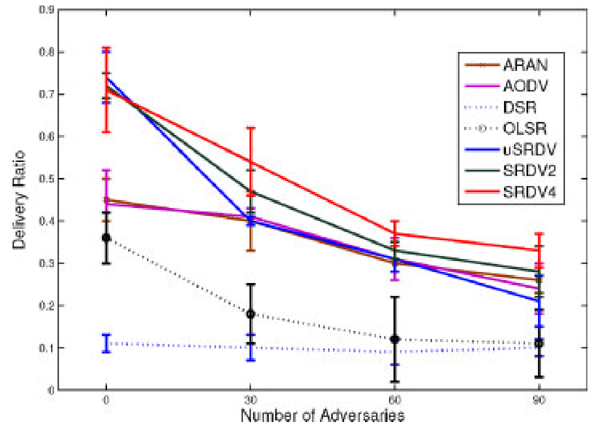

Fig. 7. Delivery versus adversaries.

as the number of delivered packets converge as seen in Figure 7. With too many attackers, the only packets that gets delivered are the ones where the source and destination are one or two hops apart and this would be true for most protocols. Nonetheless, SRDV still outperforms the other protocols in terms of packets delivered.

5.7.2. End-to-end delay

As the number of adversaries increase, the average end-to-end delay for all the protocols tested decreases as well as shown in Figure 8. The main reason for this is that fewer packets are being delivered and those which do get delivered travel very hops from the source to the destination. The packets which need to travel longer paths are more likely to encounter an adversary and get dropped. The average end-to-end delay for the variations of SRDV are very close together. but much better than that experienced by

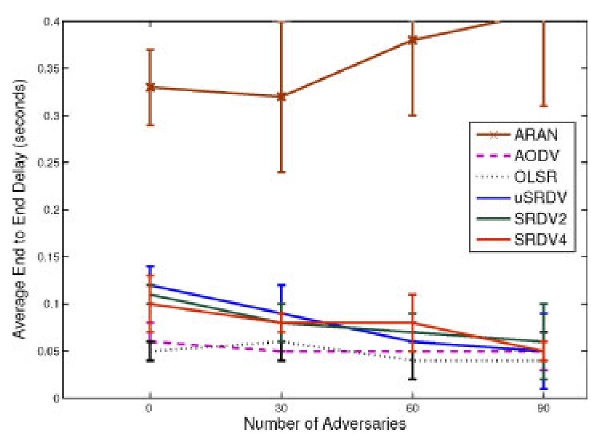

Fig. 8. Delay versus adversaries.

DOI: 10.1002/sec

ARAN. The cause of this difference is once again the susceptibility of single path protocols such as ARAN to adversaries which drop route reply messages.

5.7.3. Overhead

The general trend experienced by the protocols is that as the number of adversaries increases the overhead experienced by the network decreases. This decrease in overhead is not associated with an increase in performance but rather an increasing number of attacks which are not detected. With no adversaries, ARAN requires significantly more overhead than the other protocols. Each route discovery process causes route request packets to be flooded throughout the network. SRDV sets up multiple paths and proactively maintains them, therefore the route discovery process happens on a less frequent basis with SRDV than ARAN. As the network gets larger, the effect for flooding becomes more severe hence the big difference between the experiments with 100 nodes and those with 200 nodes (when compared to Table IV). The overhead experienced by OLSR remained almost constant as the number of adversaries increases and this is expected of proactive protocols (the actual value was around 54 ± 5 and is beyond the scale of Figure 9.

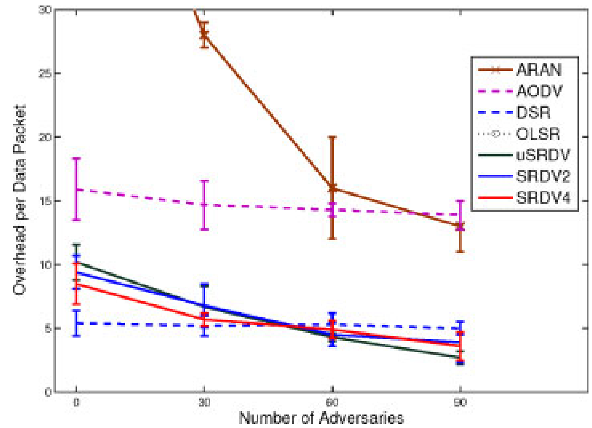

Fig. 9. Overhead versus adversaries.

5.8. Performance with Colluding Adversaries

One form of attack that has received significant attention lately is wormhole attacks [1]. To demonstrate that SRDV is capable of detecting and defending against this attack, we compare the protocols in a simulation scenario in which the network is subject only to these wormhole attacks. Therefore, the success or failure of the protocols can be attributed to their effectiveness against wormhole attacks. Of the 100 nodes in the network, we select five pairs randomly, with each of the corresponding 10 nodes being different, and connect the members of each pair with a wired link. This link is used to *tunnel* control packets from one point to the other, and represents a worst-case scenario for SRDV, because wormhole attacks using paths among colluding attackers can be detected by the end-to-end measurements taken by SRDV. All 10 of the attacker nodes then drop all the data packets they receive. This form of wormhole attacks cannot be detected without end-to-end feedback, and once they are detected, choosing alternate paths is the solution. The results of the experiments are shown in Table VI. By comparing these results to those with no attacks, we can see that the simulated wormholes do present a threat, demonstrated by the reduced performance of AODV. However, the performance of SRDV with or without wormholes is almost identical, which demonstrates its effectiveness in terms of dealing with wormhole attacks.

Given that there were no other forms of attack, the performance difference between SRDV and uSRDV can be attributed to the manner in which SRDV deals with wormhole attacks. It is evident that the use of end-to-end feedback and path diversity helps improve routing in the face of wormhole attacks. The detection process requires a significant loss of packets in order to prevent premature behavior. Both SRDV and uSRDV

Table VI. Simulation results: wormhole attacks.

	Scenario A			Scenario B		
	Delivery ratio	Latency	Net load	Delivery ratio	Latency	Net load
aAODV	0.45 ± 0.08	0.05 ± 0.02	13.4 ± 9.5	0.63 ± 0.08	0.057 ± 0.03	6.6 ± 1.4
DSR	0.13 ± 0.14	0.92 ± 0.71	8.7 ± 1.3	0.13 ± 0.12	1.02 ± 0.8	6.6 ± 2.1
ARAN	0.53 ± 0.09	0.23 ± 0.07	22.1 ± 3	0.87 ± 0.07	0.14 ± 0.08	8.9 ± 4.1
uSRDV	0.68 ± 0.11	0.127 ± 0.048	10.0 ± 1.6	0.96 ± 0.03	0.04 ± 0.01	3.2 ± 0.7
SRDV2	0.70 ± 0.06	0.07 ± 0.03	8.2 ± 1.5	0.88 ± 0.05	0.05 ± 0.02	4.0 ± 0.3
SRDV4	0.71 ± 0.05	0.09 ± 0.03	7.2 ± 0.67	0.88 ± 0.03	0.05 ± 0.02	4.9 ± 0.4

DOI: 10.1002/sec

outperform aAODV and ARAN in both scenarios. ARAN is almost immune to wormhole attacks because its ordering is dependent on time and this cannot be fabricated by adversaries, so long as the wormhole is of comparable speed to an actual multi-hop path. Immunity to one particular attack does not justify the comparatively poor performance. In Scenario B, because of the smaller network diameter, wormhole attacks have reduced effectiveness and this is reflected in the results.

5.9. The Impact of Path Diversity on Security

We have presented results from a multitude of scenarios that show side by side the performance of SRDV with one path (uSRDV), with two paths (SRDV2) and with four paths (SRDV4). Using multiple paths simultaneously can have both positive and negative consequences. As the number of paths increases, the probability that one of these paths will break or will include an adversary will increase and this can cause packets to be lost. On the other hand, by comparing the performance of different paths, attacks can be quickly detected and malicious nodes can be avoided by sending most packets though the path with better performance. By itself, path diversity is an ineffective solution however, it becomes a powerful tool when used in conjunction with load balancing based on end-to-end feedback as was done in the SRDV protocols. The simulation results show, that in general, the performance increases as the number of paths increase. Surely there will be a limit to this, and it will depend on the connectivity of the network, and in particular the number of neighbors a node have. If there are more paths than neighbors, some paths will be used for different labels and there will be no further gains in improvement from this.

6. Conclusions

We have argued that previous solutions for securing routing in MANETs have significant limitations, and presented SRDV as an instantiation of an approach based on end-to-end verification of path characteristics and the use of path diversity. SRDV implements on-demand routing through multiple paths integrated with end-to-end probing of network performance. SRDV addresses all of the security problems identified with prior approaches for secure routing in MANETs. We also illustrated through simulation experiments that SRDV is at least as efficient as traditional MANET routing protocols (e.g., AODV, DSR, OLSR) in the absence of attacks, and that it attains better performance under attacks than protocols that simply rely on single-path routing and the authentication of control packets.

References

1. Hu Y-C, Perrig A, Johnson DB. Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications* 2006; **24**(2): 370–380.
2. Hu Y-C, Johnson DB, Perrig A. Sead: secure and efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks* 2003; **1**: 175–192.
3. Ericksson J, Faloutsos M, Krishnamurthy SV. Routing amid colluding attackers. In *Proceedings International Conference on Network Protocols (ICNP'07)*, 2007.
4. Sanzgiri K, Dahill B, Levine BN, Shields C, Belding-Beyer E. A secure routing protocol for ad hoc networks. In *Proceedings International Conference on Network Protocols*, 2002.
5. Perkins CE, Bhagwat P. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. In *Proceedings SIGCOMM'94*, August 1994; 234–244.
6. Hu Y-C, Perrig A, Johnson DB. Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks* 2005; **11**: 21–38.
7. Johnson DB, Maltz DA, Broch J. Dsr: the dynamic source routing protocol for multi-hop wireless ad hoc networks. In *Ad Hoc Networking* (Chapter 5), Perkins CE (ed.). Addison-Wesley, NJ, USA, 2001; 139–172.
8. Murphy SL. Digital signature protection of the ospf routing protocol. In *Proceedings Symposium on Network and Distributed System Security*, February 1996.
9. Smith BR, Murthy S, Garcia-Luna-Aceves J. Securing distance-vector routing protocols. In *Proceedings Symposium on Network and Distributed System Security*, February 1997.
10. Perlman R. Network layer protocols with byzantine robustness. *PhD Thesis*, Massachusetts Institute of Technology, October 1988.
11. Bolo J-C. End-to-end packet delay and loss behavior in the internet. In *ACM SIGCOMM*, 1993.
12. Keshav S. A control-theoric approach to flow control. In *ACM SIGCOMM*, 1991.
13. Prasad R, Dovrolis C, Murray M, Claffy K. Bandwidth estimation: metrics, measurement techniques, and tools. *IEEE Network*, 2003; **17**: 27–35.
14. Melander B, Bjorkman M, Gunningberg P. A new end-to-end probing and analysis method for estimating bandwidth bottlenecks. In *Global Telecommunications Conference*, 2000.
15. Kang SR, Liu X, Loguinov MDD. Packet-pair bandwidth estimation: Stochastic analysis of a single congested node. In *IEEE Internation Conference on Network Protocols*, 2004.
16. Dovrolis C, Ramanathan P, Moore D. What do packet dispersion techniques measure. In *IEEE Infocom*, 2001.
17. Burrows M, Abadi M, Needham R. A logic of authentication. *ACM Transactions on Computer Systems* 1990; **8**(1): 18–36.
18. Haas Z, Pearlman M. The zone routing protocol (ZRP) for ad hoc networks. *IETF Internet draft*, 2002.
19. Kurkowski S, Camp T, Navidi W. Minimal standards for rigorous MANET routing protocol evaluation. *Technical Report MCS 06-02*, Colorado School of Mines, 2006.

DOI: 10.1002/sec