Title

Secure Routing in MANETs Using Local Times

Permalink

https://escholarship.org/uc/item/6974r4w2

Author

Garcia-Luna-Aceves, J.J.

Publication Date

2012-04-11

Peer reviewed

Wireless Netw (2012) 18:811–826
DOI 10.1007/s11276-012-0435-2

Secure routing in MANETs using local times

Stephen Dabideen · J. J. Garcia-Luna-Aceves

Published online: 11 April 2012

Abstract We present a new approach to secure routing in mobile ad-hoc networks based solely on the relative transmission times of overhead packets. Unlike most previous works aimed at securing route computation, we eliminate a key vulnerability (explicitly stated routing metrics) altogether. We introduce the Secure Time-Ordered routing Protocol (STOP), which uses time-based orderings to ensure the establishment of multiple loop-free paths between a source and a destination. STOP is the first routing protocol to use performance-based path selection without source routing, path vectors, or complete topology information, making it far more efficient that similar approaches. We prove that adversaries cannot take any action to manipulate the time-based ordering so as to unfairly gain control of the forwarding topology and, by design, nodes which drop data packets will be avoided. Furthermore, at convergence, traffic load is evenly distributed over the well-performing paths, so adversaries cannot gain complete control over the data flow through temporary good behavior. Simulation results show that the countermeasures in STOP are effective against a variety of attacks from independent and colluding adversaries, and that this improved security does not come at the expense of routing performance.

S. Dabideen (✉) · J. J. Garcia-Luna-Aceves
Raytheon BBN Technologies, 10 Moulton Street, Cambridge, MA 02138, USA
e-mail: dabideen@bbn.com

J. J. Garcia-Luna-Aceves
Department of Computer Engineering, University of California, Santa Cruz, 1156 High St., Santa Cruz, CA 95064, USA
e-mail: jj@soe.ucsc.edu

Keywords Ad-hoc networks · Security · Routing · Time-based ordering

1 Introduction

Mobile ad-hoc networks (MANETs) are susceptible to a variety of attacks aimed at preventing the delivery of data packets. A core vulnerability is the deterministic manner in which routing decisions are made and the difficult task of securing a distributively determined routing metric amidst colluding adversaries. Most routing approaches favor the use of shortest paths; therefore, to gain control of the forwarding topology an adversary can either be on the shortest path to a destination, or manipulate the route computation so that it appears to be on the shortest path. The adversary then gains control of the forwarding topology allowing it to perform denial of service or disclosure attacks. Advertising false topology information is one of the simplest attacks to a routing infrastructure and remains one of the most difficult to prevent in MANETs, especially amidst colluding adversaries. Ad-hoc networks are particularly vulnerable because the ordering is established distributively and it is difficult to verify the accuracy of the advertised connectivity. Our survey of related work in Sect. 3 indicates that most previous work is aimed at preventing the manipulation of route computation by attempting to secure the routing metric. However, most of these approaches are still vulnerable to colluding adversaries and those mechanisms that do provide security against colluding adversaries come at the cost of much larger complexity or specialized hardware (e.g., requiring time synchronization or GPS devices for packet leashes [13]).

The approach taken in this paper is fundamentally different, far simpler and yet more effective than previous approaches. Section 4 discusses the time-based ordering approach first introduced in [6] and its inherent properties, which we leverage to deliver secure routing in MANETs. Section 5 presents the Secure Time-Ordered Protocol (STOP), which constitutes the first routing protocol for MANETs that can be proven to be secure without the use of verifiable updates regarding distances, link states or path vectors. STOP is also the only routing protocol to use performance-based path selection of multiple paths without complete topology information.

Many malicious attacks are aimed at forcing data to be routed through an adversary, and this is achieved by manipulating the route computation (e.g., changing the hop count or path vector) so that the adversary appears to be on the best path to the destination. Despite several attempts [12, 14] the routing metric cannot be completely secured especially against colluding adversaries. As long as intermediate nodes must explicitly state their location, either as a the number of hops to the destination, path to the destination or co-ordinates, adversaries can misrepresent their position. This is the fundamental security limitation of most routing protocols proposed to date. STOP eliminates the need to state explicitly a routing metric to ensure correct routing, and instead uses a time-based ordering. This makes it impossible for adversaries to manipulate route computation in their favor, even with collusion as we discuss later.

STOP establishes and maintains a directed acyclic graph (DAG) with multiple, loop-free paths between a source and its destination. All available paths are used to route data and path selection is done distributively. Nodes route packets through each of their successors in proportion to their past performance, as determined by the feedback given by the destination. The manner in which we distribute traffic load is unique in many ways. As long as nodes take corrective steps to improve performance, they are entrusted with more data, until their load equals that of the other paths with proven performance. However, temporary good behavior only gives adversaries partial access to data flows, as long as there are other paths with good performance and STOP is quick to react to poor delivery. In short, STOP is provides superior security because its routing computation is more difficult to distort or disrupt and the use of multiple paths based on feedback facilitates the detection and avoidance of adversaries.

Section 6 provides a security analysis of STOP. We argue that STOP is innately immune to a variety of attacks, does not introduce any new vulnerabilities and the remaining vulnerabilities can be countered by well known security paradigms in a manner similar to spatial orderings. We consider a variety of attacks (including fabrication, modification, deletion, rushing, black-hole and wormhole attacks) and compare the effectiveness and complexity of the countermeasures employed in STOP to those used in other secure routing protocols. We discuss why adversaries cannot prevent route discovery, manipulate route computation or drop packets without detection and correction. In short, the optimal strategy for adversaries in STOP becomes forwarding the data packets on paths to the destination.

Section 7 presents the results of simulation experiments, which indicate that, in the absence of adversaries, STOP attains significantly better performance than traditional nonsecure MANET routing protocols (e.g., AODV, DSR, OLSR). We also compare the performance of STOP to ARAN [23] and SRDV [7], which are secured routing protocols, in the presence of a variety of attacks. The results also show that STOP is better able to deal with these attacks, including wormholes [15] and rushing [27] attacks.

2 Security assumptions and attack model

In this paper we are primarily concerned with attacks that prevent a source node from successfully delivering packet to its destination. This can be done by preventing route discovery or dropping data packets. Adversaries, working independently or in collusion with other nodes, can attempt to gain control of the forwarding topology by manipulating the route computation in their favor, strategically positioning themselves in the network or by co-incidence. Colluding nodes may utilize network resources not available to other nodes such as high-speed connections between them. Adversaries may adjust their behavior at any given time.

We pay particular attention to adversaries gaining control of the forwarding paths and subsequently dropping data packets entrusted to them. For example, consider Fig. 1 where nodes are connected in a manhattan grid topology with the intention of routing packets from S to D. If node C is an adversary and advertises a hop count of 1 (although it is actually 3 hops from D), the result would be a distorted DAG, with respect to D in which all paths from S to D passes through C. If all the nodes were well behaved, the direction of

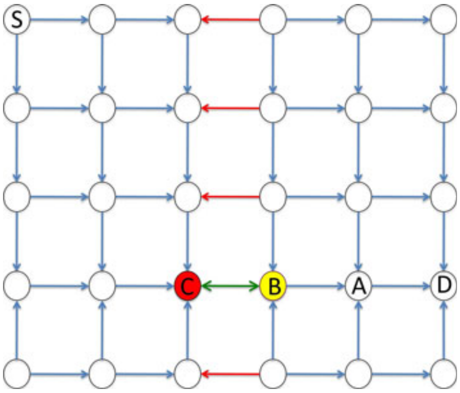

Fig. 1 Example of an adversary distorting the ordering in a network

the red arrows in Fig. 1 would be reversed and there would be many paths from S to D that did not involve C. Section 3 presents a summary of prior approaches to constructing secured orderings in MANETs, most of which focus on securing the routing metric at the cost of computational complexity. Interestingly, most of the approaches fail in the presence of colluding adversaries. For example, in Fig. 1, node B can tunnel unmodified packets it overhears from node A to node C. Node C can then retransmit the unmodified packet, with a hop count of 1 and a valid signature and hash value by pretending to be A. The only common neighbor of A and C is B which is colluding with A so this simple attack will be nearly impossible to detect and result in C gaining control of all paths from S to D. Even the use of multiple paths on this compromised DAG will not improve performance since all paths go through the adversary.

We acknowledge there are many more types of attacks than those addressed in this paper, and that the approach taken in this paper need not address all possible attacks. However, attacks that are not addressed specifically in STOP could be addressed using mechanisms similar to those proposed in the past in the context of spatial orderings. We make two assumptions regarding the network:

1. The source and destination are not adversaries.
2. There is a path without any adversaries between the source and the destination at every instant. Otherwise, it would be impossible to secure the routing process.

3 Related work

Most previous work on secure routing for MANETs relies on mechanisms that compromise scalability or performance of the routing protocol. Some previous work has attempted to secure the routing metric and make it difficult for adversaries to advertise false, short routes. Hu et al. [12] propose the *Secure Efficient Ad hoc Distance vector* protocol (SEAD) as an enhancement of the *Destination-Sequenced Distance-Vector* (DSDV) [21] routing protocol [21]. SEAD uses a hash chain in an attempt to secure the distance metric by making it difficult for adversaries to decrease the value of the routing metric. The *Ariadne* [14] protocol as an enhancement of the *Dynamic Source Routing* (DSR) protocol [16], which allows source nodes to authenticate the path, but requires multilayered cryptography in which each node on the path encrypts and decrypts a packet that is encrypted by each previous hop. These protocols are still vulnerable to colluding adversaries, which can make the path appear shorter than it actually is, without detection.

Other approaches are aimed at detecting, rather than preventing, the manipulation of the routing metric. The *Wormhole Attack Prevention (WAP)* [4] protocol, nodes compare the average transmission time between hops to the number of hops to determine of the advertised distance is shorter than the actual distance. In the *Geographical Secure Path Routing (GSPR)* [20] protocol, nodes use a novel geographic hashes that are un-spoofable to detect when a node is misrepresenting its location.

Ericksson et al. [10] proposed the *Secure Probabilistic Routing* (Sprout) protocol, 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. This approach is not restricted to shortest path routes and is indeed resilient to colluding adversaries. However, it requires each node to maintain complete topology information at all times, which may not be practical in a MANET.

A different approach to security in MANETs is controlling the dissemination of information by the assignment of *trust* levels to nodes. In SAR [26], a trust hierarchy is established and the dissemination of packets is restricted to nodes with some minimum trust level specified by the source. The main issue with this approach is determining the trust level to which each node should belong. If it is preassigned, adversaries may be assigned a high trust level or fabricate the level of trust needed. In Watchdog [18], nodes promiscuously listen to the transmission of their neighbors to determine whether or not they are forwarding packets in order to determine their level of trust. However, an adversary forwarding a packet with inaccurate information can potentially cause greater harm than if the adversary were to drop the signaling packets. While recent work [5] advocates combining trust with cryptography for improved performance, accurately gauging trust remains a daunting task. In the worse case, an adversary can be well-behaved long enough to establish a good reputation and then exploit the trust it earned. Detection will not be instantaneous and significant harm can be done before malicious behavior is detected, if it is detected at all.

Sanzgiri et al. [23] proposed the *Authenticated Routing for Ad hoc Networks* (ARAN) protocol. ARAN is an on-demand routing protocol that extends AODV and 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 secure the routing function. The strength of ARAN is that it is a simple protocol that ensures the authenticity and integrity of routing messages, and uses the fastest path traveled by the route request, which is unspoofable. Its simplicity makes ARAN almost invulnerable to distortion attacks, but it is still vulnerable to attacks where colluding adversaries can use additional resources to perform rushing attacks. Furthermore, if an adversary lies on the quickest path by luck or deliberate placement, ARAN provides no defense.

A variety of attacks have been studied in the context of routing in MANETs. In a rushing attack [27], an adversary attempts to manipulate the route computation so that it lies on most if not all of the paths discovered to gain control of the forwarding topology by speeding up the retransmission of overhead packets or through the use of increased radio range. Protocols that establish a single path, such as AODV and its derivatives, are particularly vulnerable. Wormhole attacks [15] require colluding nodes to tunnel packets from one point in a network to another and can be used to perform rushing attacks as well as other attacks. In a black-hole attack [8] nodes responds to route requests with route replies even if they do not have a path, and this results in data being forwarded to them. In a gray-hole attack [25], adversaries are initially well-behaved but eventually start performing attacks and this can be used to counter trust-based schemes.

4 Time-based orderings

We use a time-based ordering [6] to construct DAGs with respect to destinations of interest on-demand. In this approach, a node classifies each of its neighbors as a potential successor, predecessor, or neutral with respect to the destination based on the relative times when the node receives and relays route requests (RREQs). Only the relative time is of importance so clock synchronization is not needed.

Definition 1 Node A is a *successor* of Node B to destination C if $t_B^A > t_B^B + \delta$ or if A is the destination, where t_B^A is the local time node B received a RREQ from A and t_B^B is the local time at which node B retransmitted the RREQ.

If A is a successor of B with respect to C, then we can say B is A's predecessor with respect to C. If neither is true, i.e. B transmitted and received the RREQ for C from A within δ, then A and B are neutral with respect to C. The loop-free properties as well as the value of δ has been thoroughly discussed in the past [6]. In this paper, we focus on the security implications of time-based ordering.

A time-based ordering allows the creation of DAGs in which there are many paths between a source node and its destination. As there is no explicit notion of distance, it is possible to route packets over paths that need not be the shortest paths; however, this does not create routing loops. In-fact, by simply controlling the retransmission delay of RREQs a node can adjust the number of successors and predecessors it has for a given destination so as to maximize the number of paths to the destination. The DAG is built based on the relative transmission and reception times of overhead packets, as determined by the local clock of each node, so there is no need for explicitly stated metrics such as hop count or link state in the overhead packets. An important implication of this is that intermediate nodes do not need to modify any signaling packets. In the following sections, we argue that multiple paths are necessary for secure routing and DAGs based on time based orderings are therefore better that DAGs which are restricted to paths of length of the shortest path only.

5 The Secure Temporally Ordered routing Protocol (STOP)

The goal of STOP is to make *inaction* the optimal strategy for any adversary wishing to gain control of the forwarding topology. Adversaries should not be able to manipulate the route computation by physically or symbolically placing itself on the *shortest path* since there is neither a routing metric to manipulate nor any notion of *shortest paths*. The approach requires low computational and storage complexity, and intermediate nodes have minimal responsibility in the signaling of the protocol and do not modify any information in signaling packets.

STOP is designed around three key ideas: time-based ordering, performance-based path selection, and feedback from the destination. Time-based ordering is used to construct a DAG that is less susceptible to manipulation, performance-based path selection is used thwart actions of adversaries, and feedback is a corrective countermeasure used to reinforce the path-selection process.

For each active destination, a node must store: the latest sequence number for that destination, the time at which the node received the RREQ /RREP from each of its neighbors with the latest sequence number and the time at which the node transmitted the latest RREQ. Each node also maintains a 128-bit vector and current forwarding probability for each (successor, source, destination) tuple as well as an integer corresponding to the start of the window denoted W_{start} for each (source,destination) pair.

5.1 Route discovery and maintenance

Route discovery in STOP has two phases. In the first phase RREQs are initiated by the source of a data flow, and are flooded throughout the network. A RREQ consists of three fields: a destination address, a source address and a sequence number. Upon receiving a new RREQ, a node records the time of reception (according to its local clock) and the sequence number. If the node is not the destination and the RREQ is new, the *unmodified* RREQ is rebroadcasted after a small calculated delay (Δ). The value of Δ is calculated based on the previous number of successors and predecessors with the intention of maximizing the number of paths [6].

If a node receives a RREQ and is named as the destination, the node issues a RREP. RREPs contain five fields: destination address, destination sequence number, source address, source sequence number, and a 64-bit vector for feedback information.

The Reply Acceptance Condition (RAC): A node can only accept and process a RREP if it is received from a *successor*.

RREPs are retransmitted the first time they are *accepted*, as defined by *RAC*, except by the source which always accepts the RREP and never retransmits it. For the duration of a data flow, destination nodes proactively initiate RREPs every 20 s. These proactive RREPs serve to update the ordering and deliver feedback information. Although nodes may not *accept* a RREP from a predecessor, it can process the feedback information. When a node issues a RREQ it sets a timer. If this timer expires and the node is yet to receive a RREP, it increases its sequence number and issues a new RREQ.

For example, consider Fig. 2 where *S* wants to send data to *D* and *X* and *Y* are adversaries. *S* will flood RREQs throughout the network as illustrated with the arrows. *D* will subsequently initiate a RREP but it will only propagate to a subset of the network as it is limited by RAC. Nodes such as *E* and *F* will never receive a RREP from a successor. Only the links in orange will be validated and used for routing.

Time-based ordering allows a node to have multiple successors to the destination of interest. These paths are not restricted to only those of the shortest length (e.g., in Fig. 2 there are paths of length 5-hops through 7-hops in the established DAG). If one of the links fails a node can route through any of its remaining successors. If a node no longer has a path to the destination, either because of link failure or after receiving a route error message (RERR) from its last remaining successor to that destination, it issues a RERR. This RERR will serve to prevent future data packets from being routed through this node to the destination. If the source no longer has a successor to the destination it initiates a new RREQ.

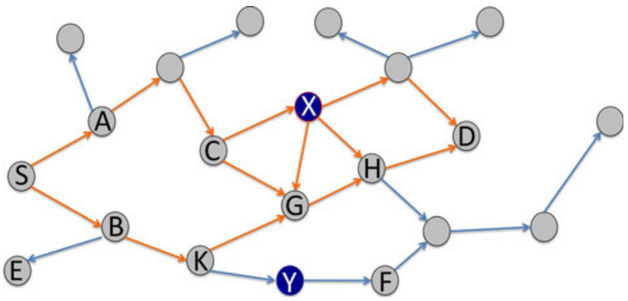

Fig. 2 An example showing the propagation of overhead packets in a time-based route computation in STOP

5.2 Performance feedback

Each data packet is labeled with an packet identifier (PID) set by the source. Nodes maintain a 128 bit-vector for each (successor, source, destination) tuple. This serves as a sliding window representing the last 128 data packets sent by the source, as known to each intermediate node. When a node receives a data packet it shifts the relevant window for each successor, inserting 0s in the bit-vectors, until the most significant bit corresponds to the PID of the data packet. The node then sets the most significant bit to 1 in the window corresponding to successor to which the packet is forwarded. For example, if node *A* has a current window for packets 100–228 (i.e. $W_{start} = 100$) and it receives a data packet with PID 235, it will first shift the window by 7 to remove the least significant 7 bits and insert 0's as the most significant 7 bits. *A* will increase the value of W_{start} to 107. If *A* chooses to forward the packet to *B* it changes the most significant bit in the packet window for *B* to 1.

Each destination includes a 64 bit-vector representing the last 64 data packets as well as a PID to identify the earliest packet for which this feedback information applies. The bit value is set to 1 if the corresponding packet was received else it is set to 0. Upon receiving this feedback information, a node performs a logical *AND* between the corresponding bits in the local window and the feedback bit vector. The sum of the bit-vector following the *AND* operation gives the number of packets delivered through that successor in the current window.

The size of the bit vectors used to record packets seen and to carry feedback information should depend on the rate at which packets are transmitted and the interval at which feedback is sent. These values should be chosen so that when a node received feedback from the destination, the local sliding window should not have gone past the corresponding packets. Also, packets may be sent while the feedback information is propagating, so it is important that the local sliding window is larger than the window in the feedback. It is not necessary to receive feedback from every packet, especially if the data is high, but the more feedback a node gets, the more accurate its performance estimates will be. In the simulations we use window sizes of 128 bits for local memory, 64 bits for feedback together with an feedback interval of 20 s but alternate parameters may be necessary depending on the application. Clearly, a smaller feedback interval will increase the network overhead, especially since this information must be propagated over the entire network. If the data rate is high, then larger windows can be used to keep the feedback interval large.

Let P_X^{AD} denote the current probability of node *X* routing a data packet to *D* through neighbor *A*. Let λ_X^{AD} denote the fraction of packets node *X* routed through *A* to *D* which were received by *D* in the current window. At initialization, and whenever there is no applicable feedback information, P_X^{AD} is assigned a value of 0.5 and λ_X^{AD} is assigned a value of 1. Upon receiving feedback information, nodes recalculate the forwarding probability for each of its successors to D according to the following formula:

$$P_X^{AD} = \left(\frac{\sqrt{P_X^{AD}} \times \left(\lambda_X^{AD}\right)^2}{\sum_{allY}\left(\sqrt{P_X^{YD}} \times \left(\lambda_X^{YD}\right)^2\right)} \right)$$

This takes into account past performance as well as performance in the current window. The use of $\sqrt{P}$ allows nodes to subsequently increase P by improving its performance and not just relative performance (this will not be true if we used P^{α} with $\alpha \geq 1$). The use of λ^2 allows for quicker and more pronounced reaction to changes in performance. If there are multiple paths performing well, in steady state, the majority of packets will be distributed among these paths in proportion to their performance.

5.3 Data forwarding

In STOP, a data packet is routed through randomly selected successors at each relay, until it arrives at the destination. Each node maintains a list of successors with respect to a destination. Each node assigns the integer between 0 and 99 to its successors, where the number of integers assigned to any node is proportional the relative performance of the paths from that successor to the destination. When a node receives a data packet, it picks a random number and the node which is assigned that random number is used to forward the data packet. Therefore, the probability of any particular successor being chosen to forward a data packet as based on past performance, as determined by the destination's feedback. STOP does not attempt to identify adversaries, which is far more difficult, but rather attempts to avoid ill performing paths. Eventually, STOP converges to the use of multiple paths with verified performance, provided that there are multiple paths which successfully deliver packets.

The deterministic nature in which routes are chosen is a critical vulnerability of many previous approaches to secure routing. For example, if a routing protocol chooses shortest path then an adversary's optimal strategy is to physically place itself on the shortest path or merely manipulate the route computation so that an adversary appears to be on the shortest path. With randomized path selection, there is no optimal strategy for adversaries. However, randomized path selection by itself is insufficient. If an adversary happens to be on the randomly chosen path, it can drop packets as long as corrective measures are not taken. In STOP, the path selection is initially random, but becomes performance-based once feedback information is available. If an adversary happens to be on the randomly chosen path and it drops packets, fewer packets which be routed through it. The optimal strategy in in STOP is for adversaries to forward more data packets on paths to the destination than the other routes, which is not in itself an attack. If there are paths with better performance, after each successive update, fewer packets will be routed through paths containing packet-dropping adversaries.

In Fig. 2, S initially routes half of its data through A and half through B. If X drops data packets then C will detect that the path though X is less reliable than the path through G and will route a greater fraction of its data through G. The source S will also observe that the path through B is more reliable than the path through A and will route a greater fraction of packets through B. The performance of the path through A will improve as C sends less data through X and S will increase the fraction of packets routed through A. If the network is stationary, at convergence the path S–B–K–G–H–D and S–A–C–G–H–D will be used to route almost all the data with each being used equally. The time to convergence depends on the behavior of the adversary. If X drops all data packet, then after the first feedback, $\lambda_C^{XD} = 0$ resulting in C sending all its data packets through G. (But for practicality we place a lower bound of 0.05 on the forwarding probability to allow nodes to recover if they are able to deliver the few packets entrusted to them). For simplicity, consider what would happen in Fig. 2 if X dropped half of the data packets routed through it, all data packets are delivered except those dropped by adversaries (X and Y) and that the topology does not changes. Table 1 shows various forwarding probabilities after 4 sets of feedback information.

Table 1 reveals some key features of the design of STOP. Node C reacts sharply after the first feedback and as a result of C's corrective behavior and improved subsequent performance, C regained a larger data flow from from S. C does not attempt to identify X as an attacker but it knows that X is not reliable, possibly due to an adversary somewhere on the path between X and D. The use of the square root function results in P_C^{XD} slowly approaching 0 but C continues to route few packets through X allowing X to take corrective behavior and then increase the

Table 1 Forwarding probabilities of nodes S and C

Iteration	P_C^{XD}	P_C^{GD}	P_S^{AD}	P_S^{BD}
0	0.50	0.50	0.50	0.50
1	0.20	0.80	0.36	0.64
2	0.11	0.89	0.38	0.62
3	0.08	0.92	0.40	0.60
4	0.07	0.93	0.43	0.57

proportion of packets entrusted to it. In this example, the protocol converges to the two safe paths with approximately equal load.

5.4 Authentication

In a secure routing protocol, packets must be authenticated with respect to the identity of the origin, the identity of the sender and the contents of the packets. Without authentication, it would be very simple for an adversary to perform a variety of attacks by masquerading as different nodes, especially as the destination.

To date, most work in the area of authentication has focused on verifying the identity of the source of the packet and the node retransmitting the packet. Key distribution and management is essential in a secure routing protocol, including STOP. However, this area is well studied and it is not the aim of the paper to propose any novel key distribution or management techniques. Instead, we assume the existence of a cryptosystem running independently of, and in parallel with, STOP to distributively issue and revoke cryptographic keys. Many such systems have been designed specifically to meet the constraints of MANETs [2, 9] and can be applied to STOP. We assume that, as long as the cryptosystem is not compromised, adversaries cannot replicate the signature of a legitimate node on an arbitrary packet. The computational complexity and overhead proposed cryptosystems vary and the separation of routing and key management allows for the most suitable cryptosystem to be applied to the specific network. All packets are also signed by the transmitting node to make impersonation attacks more difficult, but the local signature is not propagated only the signature of the origin is propagated. For example, consider Fig. 2, where node S initiates a RREQ and it is retransmitted by nodes B and K in that order. Let $E_X(Y)$ denote the result of encrypting packet Y with $X's$ key. Node S will transmit $E_S(RREQ_S)$, node B will receive $E_S(RREQ_S)$ and retransmit $E_B(E_S(RREQ_S))$, node K will receive $E_B(E_S(RREQ_S))$ and retransmit $E_K(E_S(RREQ_S))$. Each packet is signed at most two times, to authenticate the origin of the packet and the previous hop. This level of authentication is not novel and has been previously applied to routing in wireless networks.

One advantage of STOP is the ease of authenticating the contents of a packet. Most approaches to routing in MANETs rely on the use of some explicitly stated routing metric such as hop count, link state or path vectors. Intermediate nodes are responsible for updating the routing metric used and therein lies their weakness; this field cannot be protected by the source. As long as intermediate nodes are responsible for modifying part of a packet, adversaries can insert false information, such as a smaller hop count, which is difficult if not impossible to authenticate. The origin of packet can be identified by cryptography but it is more difficult to verify the routing metric stated in the packets. Even with hash chains, as we explain in the next section, adversaries can advertise hop counts to an arbitrary destination that is smaller than the actual number of hops to that destination.

In STOP, there is no explicitly stated routing metric and intermediate nodes do not modify in any way the contents of packets. If S is the source of a packet, $E_S(Packet)$ is always retransmitted by intermediate nodes and any modification would be detected upon decryption. In STOP, the routing metric is the reception times of packets which is implicit and the performance of paths paths which can be secured by the destination. Provided that some secure key management scheme in place, the origin, the previous hop and the complete contents of packets can all be authenticated in STOP.

5.5 Applications of STOP

Some of the design choices in STOP impacts the types of applications which this protocol can support. In STOP, packets traverse multiple paths and can therefore arrive out of order. Also, nodes introduce some small, but finite delay when forwarding packets. As presented, STOP cannot support applications which are not very sensitive to variations in delay and which can handle out of order packets. One example of an application which STOP can support is TIGR [11], where nodes capture and transfer media files. Fragments of the file can arrive out of order and reassembled at the destination. With some adjustments, STOP might be adapted to applications such as VoIP, which requires packets to arrive in order. In this case, the data stream can be broken up into chunks of sequential packets of arbitrary size and each chunk is transmitted along a single path. The size of the chunk should be in proportion to the performance of the path.

STOP would be better suited to applications which requires communication over a longer period of time rather than short bursts. The forwarding probability is based on the relative performance of the paths and it takes several updates before the nodes have a meaningful measure of the performance of the paths through all their successors. In STOP, initially all paths are selected with equal probability which can lead to poor performance at first. However, with each feedback update, less data is routed through adversaries and leading to a higher proportion of delivered packets.

5.6 Scalability of STOP

The design choices have some positive and negative impacts on the scalability of STOP. The encryption used to provide authentication in minimal. Each packet is signed at most twice: once by the origin and once by the node forwarding the packet. When a packet is retransmitted, the encryption of the previous hop is removed unless the previous hop is the origin. Some protocols [14] require each node to sign the packet and maintain the signature of every other node in the path. As the network size increases, the time and processing required hinders the scalability. This is clearly not the case in STOP.

In a single route computation, multiple paths are established. When links break, due to mobility, node failure or any other reason, an alternate route can be used. This approach reduces the frequency at which route computation must be performed and can therefore significantly reduce the overhead of the protocol, improving its scalability.

The frequency and size of the periodic RREPs can limit the scalability if the parameters are not chosen carefully. Feedback information must be propagated through all nodes in the network to mitigate the effects of adversaries deleting RREPs. As the network gets larger and the frequency of updates get smaller, the overhead involved in sending the feedback information can hinder the scalability of STOP. A smaller update frequency can used with larger windows to have the same effect but less, although larger, overhead packets.

6 Security analysis

STOP attempts to ensure that an attacker cannot disrupt or manipulate the route computation. Manipulation of the routing computation allows an attacker to control the forwarding paths. 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. In the following, we identify possible attacks on the routing protocol, characterize threats posed by these attacks, describe the countermeasures implemented in STOP to eliminate or mitigate them, and prove that these countermeasures are sufficient for secure routing in MANETs.

6.1 Fabrication attacks

An adversary can attempt to disrupt routing by fabricating RREPs to gain an advantage in the forwarding topology. Adversaries can also fabricate RERR messages, masquerading as a different neighbor, causing nodes to remove legitimate paths from their routing table, possibly forcing them to use compromised paths or causing a denial of service attack.

This attack is countered effectively in STOP, as well as many other secure routing protocols, through the use of cryptography to authenticate the source and the content of packets. In ARAN [23], nodes use a trusted certificate server to obtain public and private key pairs which they use to sign and decrypt packets. Similarly, the authors of SAR [26] propose the use of simple passwords or a trusted third party to provide authentication service. However, this is far simpler in STOP than most other protocols, because intermediate nodes do not modify signaling packets. Packets are only encrypted by the source and the one hop neighbor as no other information is relevant. Other protocols require more complex schemes, such as the use of hash chains [12] or having each node encrypt a packet already encrypted by every previous node in the path [26], to secure the routing metric. As long as there is some key management scheme in place, STOP can provide the same level of protection against fabrication attacks as other protocols.

6.2 Modification attacks

A modification attack occurs when an adversary updates a routing packet such that it conveys false information for example a shorter hop count or path vector. This can be done to gain advantage in the forwarding paths allowing denial of service attacks. One countermeasure to prevent adversaries from using incorrect distance information is the use of hash chains [12]. Hash chains are still vulnerable, because adversaries can hash the value more than once, not at all, or tunnel packets to reduce the number of time it is hashed. Attempts have been made to secure path vectors in the same manner using cryptography and hash chains [14] but this is computationally expensive and still vulnerable to attacks where the last few hops are removed or packets are tunneled to reduce the path length. A different approach [24] only requires cryptography from two hops, but requires the establishment of secured two-hop neighborhood information which is also a difficult task.

Intermediate nodes in STOP cannot modify any field of signaling packets; any modification would be detected, given that authentication would fail. Hence *STOP is immune to modification attacks*. This is significant, because *previous countermeasures to modifications are usually computationally complex and not completely secure against this attack*. In addition, while link-state protocols are such that intermediate nodes do not modify signaling packets, nodes can collude to report links that do not exist, thereby changing the topology from which nodes compute routes.

6.3 Replay attacks

In a replay attack, an adversary would transmit an old packet (RREQ, RREP, RERR) at a later time. Since the packet is not fabricated, it will pass authentication as it will carry a valid signature. Replay attacks can have the same effect as a fabrication attack.

The countermeasure to this requires cryptography and is already used in several routing protocols [14, 23, 24]: the use of a source sequence number that is signed by the source. Older packets will have outdated sequence numbers and can be easily detected. Intermediate nodes cannot alter the sequence number field, and this field should therefore be immune to modification attacks. This is the same approach used in STOP, and it allows the detection of all replay attacks.

6.4 Deletion attacks

An adversary can attempt to thwart route discovery by deleting overhead packets. Protocols which attempt to construct a single path to the destination, such as AODV and its derivatives (e.g., ARAN), are particularly vulnerable to this type of attack. If an adversary is on the path that is being established, either legitimately or via some attack, dropping this RREP can prevent route discovery, forcing the source to repeat the procedure. Without detection, the attack can be repeated on subsequent computations, resulting in a denial of service attack.

STOP sets up multiple paths and if the adversary were to delete a RREP, a path containing an adversary will not be discovered and the RREP will arrive at the source through another path, should one exist. Adversaries cannot alter the feedback information in RREPs, and if an attacker were to drop a RREP, its neighbors will either get the feedback from a different neighbor or not have a path to the destination. Thus, an adversary gains no advantage from dropping RREPs in STOP. Proactive protocols such as Sprout [10] are also immune to these attacks. The limitation with STOP, and any other routing protocol, is that if there is only one path, nothing can be done to prevent deletion attacks.

6.5 Rushing attacks

In a rushing attack [27] adversaries are assumed to have means of communicating faster than other nodes. They take advantage of the fact that only the first RREQ is retransmitted to gain control of the forwarding topology. Protocols that establish a single path are particularly vulnerable to rushing attacks. Some countermeasures are presented in [27] and include neighborhood verification and randomized message forwarding. The authors [27] described how these mechanisms can be applied to on-demand distance and path vectors. However, these mechanisms come at the cost of computational and signaling complexity.

Protocols based on distances are vulnerable to rushing attacks, given that early transmission can be viewed as the shortest path. Extended radio range can result in shorter paths lengths in AODV, shorter path vectors in Ariadne and ARAN is especially vulnerable since it relies on the quickest path. In STOP, rushing a RREQ would result in an adversary having many successors (neighbors that transmit subsequently in time) and few predecessors (neighbors that transmit before in time) and this is not an advantageous position, because attackers need predecessors to receive data packets. RREPs travel many paths and increased radio range will not disrupt the routing computation in STOP as in single path protocols.

6.6 Delaying attacks

In a time-based ordering, there is no metric nodes can manipulate and the ordering is based on transmission and reception times, and adversaries can manipulate this, but only by delaying transmissions. If an adversary were to delay the transmission of RREQs, most of its neighbors will consider the adversary a successor, because it transmitted after them. They will subsequently route some packets through this adversary; however, if the adversary were to drop data packets, this would be detected in the feedback information and nodes would send fewer packets on paths suspected of containing adversaries.

6.7 Black-hole attacks

In a black-hole attack an adversary always respond to RREQs with a RREP regardless of the existence of a path through it. If the response propagates to the source first then the adversary will be used to route data, which it can then drop. Proposed solutions [1] include nodes waiting for multiple RREPs before forwarding to ensure there is actually a route, but this approach would fail if there is collusion among the adversaries. In [8] the author use a *Further Request* to verify the next hop in the path. But this too is susceptible to collusion among successive black-hole nodes. To deal with cooperative black hole attacks, the use of a path monitoring scheme has been proposed [22] in which a node does not forward data to another node unless it has successfully forwarded to the same destination through that particular neighbor in the past.

In STOP, we take a simpler approach. Intermediate nodes are not allowed to initiate RREPs. With the use of authentication, adversaries will not be able to fabricate or replay RREPs making STOP *immune to black hole attacks*.

6.8 Wormhole attacks

In a wormhole attack [15] colluding adversaries tunnel packets through each other from one node in the network to another node. This can be difficult to detect because the adversaries impersonate legitimate nodes. In Fig. 3 nodes *A* and *B* are colluding to form a wormhole. Suppose S initiates a RREQ which arrives at *D* which then initiates a RREP. Once *B* receives the RREP from *E*, it can tunnel the RREP to *A* which can then retransmit *E*'s RREP (with *A* claiming to be *E*). *S* believes *E* is a neighbor and *E* is actually two hops from *D*, so *S* chooses to route through *A* (thinking it is *E*). Packet leashes [13] has been used to counter these attacks to some extent. In packet leashes, nodes include their geographical position or the time of transmission in packets and this allows receivers to calculate the distance of the hop and thereby detect tunnel packets. The limitation of this approach is the need for geographic location information or tight clock synchronization, which is unreasonable in large networks. The SECTOR [3] protocol allows nodes to measure the distance between neighbors by issuing a 1-bit challenge and having the neighbor respond immediately. This approach does not require clock synchronization, but requires special hardware.

Wormhole attacks can be even more malicious and even harder to detect. Consider Fig. 4. Nodes *F* and *G* tunnel packets between them using either some external link or simply routing packets through the network. If *G* pretends to be *F* or *D* by retransmitting tunneled packets the attack can be detected, but using the tunnel (which may not be actually be a single link) *G* can claim to be adjacent to *F*. The increase in delays or distances associated with tunnels will only be detectable by *F* and *G*, which are the adversaries. Therefore, they can claim to be neighbors and provide the shortest path, without detection by previous approaches.

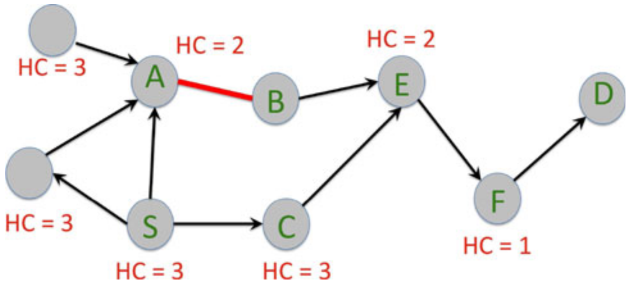

Fig. 3 A simple wormhole attack

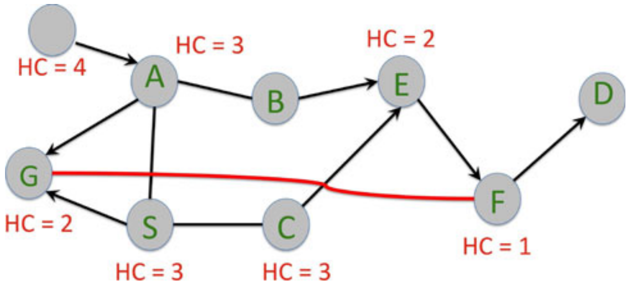

Fig. 4 An advanced wormhole attack

In the Wormhole Attack Prevention (WAP) protocol [4], nodes use transmission times to verify the hop count to detect wormhole attacks. Nodes compare the time between sending a RREQ and receiving a RREP to calculate the average time between hops. If the average time is longer than the maximum time taken to transmit the maximum range, then there must be a wormhole attack. However, the approach taken in WAP does have some limitations. If the nodes are mobile, the velocity of the nodes are needed and this requires equipment which may not be available, and the extra signaling to maintain up-to-date velocity information may impact performance. Some error must be introduced to compensate for delays in processing and retransmission delays. Also, if the nodes are not the maximum distance apart, then there can be some undetected wormholes. For example, consider a scenario where the maximum transmission range is *T*, node *A* is at a distance of *T*/2 from node *B* and node *C* is a distance slightly more than *T* and *T*/2 from nodes *A* and *B* respectively. Node *B* can pretend to be node *C* undetected as long as there is some error in timer to allow for delays. Even without the introduction of errors in the timer, wormholes can still go undetected, to some extent, as long as they do not make the path appear to be less than half the actual distance and the topology allows it. Furthermore, if the adversaries use a link with a higher transmission rate, packets can be tunneled through the wormhole, carry and smaller hop and go undetected.

STOP employs two layers of wormhole protection that requires neither extraneous equipment nor knowledge of the nodes' velocity. There is no hop count to modify so merely tunneling the packets (to avoid modification of hop count or path vectors by intermediate nodes) will not pose a threat. The equivalent to manipulating distance in distance vector protocols is manipulating time in STOP. However, since STOP uses all paths and does not favor the quickest path, there is no advantage to speeding up delivery of RREPs through the use of extraneous high speed links, Furthermore, any successful wormhole attack which is used to perform denial of service attacks by dropping data will be detected and nodes will modify their path selection based on the destination's feedback.

6.9 Attacks on data

If an adversary is used to route data it can drop some or all of the data entrusted to it. In Sprout [10], nodes maintain complete topology information and use source routing on probabilistically generated paths. Paths are selected based on performance. In SRDV [7], two paths are used together with end-to-end feedback. More packets are routed on the path with better performance and if the performance of a path does not exceed a threshold, the source forces a reordering of the network.

In STOP, we use an approach similar to Sprout, except there is no need for complete topology information or source routing. Packets are routed incrementally with each node making a local decision based on the forwarding probabilities (which reflects past performance) of its current successors. STOP should also converge faster than Sprout, because it explores all paths simultaneously and offers greater flexibility than SRDV, because more paths are used.

6.10 Limitations of STOP

STOP was designed to take advantage of multiple paths between a source and its destination. If the network topology is such that there is only on path between the source and the destination, then this path must be used to route data. If this path contains an adversary, then the feedback information would be useless as there is no corrective measure that can be taken other than not sending packets. But is true of any routing protocol if there is only one path and this lone path contains adversaries. Nonetheless, with the time-based ordering in STOP, it is more difficult for adversaries to distort the topology such that there appears to be only a single usable path to the destination. Protocols which aim to discover only a single path [23] or that use only routes of the shortest paths are more vulnerable as adversaries can advertise a lower hop count and gain control of the forwarding topology.

Theorem 1 *Adversaries, working either independently or in collusion with any number of adversaries, cannot manipulate the route computation in STOP so that more packets are routed along paths containing adversaries than paths without adversaries.*

Proof Assume for contradiction that adversaries are able to manipulate the route computation process so that paths that include them have a higher forwarding probability, after performing some form of attack, when compared to the uncompromised route computation. The fabrication, modification or replay of RREQs or RREPs would require the digital signature of the source or destination, respectively, and we assume that the source and destination are not themselves adversaries; therefore, these attacks would be detected and ignored. The only undetectable attack on the route computation would be the deletion or delay of overhead packets. If the adversary were to delete a RREQ or RREP, it would not be eligible to be a successor at any of its neighbors; therefore, this action would make it impossible for the adversary to be in a path to the destination. The timing of RREPs has no effect in the ordering whatsoever. Delaying a RREQ makes the adversary appear as a successor to its neighbors, but does not prevent them from discovering or using other paths. However, if the RREQ is delayed too long, the adversary will have no successors to route data and this will be detected in the feedback, resulting in the adversary being avoided. Therefore, the best approach an adversary can take to be on a path to the destination is to forward unmodified RREQs and RREPs within the expected timeframes and forward data packets. This contradicts the assumption that adversaries can actively manipulate route computation in their favor.

6.11 The security of STOP

We now argue that the various security mechanisms of STOP work together ensure its security. In summary, adversaries cannot prevent route discovery, cannot manipulate the route computation to gain control of the forwarding topology, and if they do happen to be entrusted with data packets which they then drop, the attack will be detected and corrective measures will reduce the number of dropped packets.

To prevent route discovery, adversaries must take action to ensure that either no uncompromised RREQ arrives at the destination or that no uncompromised RREP arrives at the source. The simple cryptography, together with the use of sequence numbers, ensures that compromised RREQs and RREPs are detected and discarded. RREQs are flooded throughout the network; hence, the RREQ must eventually arrive at the destination through at least one path with no adversaries. Likewise, there will be many paths for the RREP to travel to the source.

An adversary can manipulate the time-based ordering only to the extent that all its neighbors consider the adversary as a successor (by delaying the retransmission of the RREQ) but there is no way to become the only successor to all its neighbors. This is in stark contrast to spatial ordering, where an adversary can advertise a shorter path than all its neighbors (via wormhole, modification or fabrication attacks) to gain control of the forwarding paths. The use of time-based ordering ensures that multiple paths are established, as long as the physical topology allows.

Routing in STOP is based on performance and not a metric used to construct a DAG for routing. Nodes use all available paths in proportion to their past performance, as specified in the destination's feedback. Delivery of this feedback information cannot be prevented, because it travels many paths and it cannot be altered without detection since it is signed by the destination. In light of this, the best action an adversary can take is to forward data packets. The parameters of the protocol are selected such that the data packets are evenly distributed among all paths with good performance. Therefore, an adversary cannot gain control of all the data packets by behaving well for a period of time and then start dropping packets. At best, it can gain control for a fraction of the data packets, as long as the topology allows multiple paths. Furthermore, STOP attains this more efficiently than other security protocols.

7 Simulation results

We use simulation experiments to demonstrate that the security mechanisms in STOP is effective against a variety of attacks and this defense does not come at a cost in routing performance. In fact, the results clearly show that the multi-path routing based on time-ordering is a solid foundation, as STOP clearly outperforms the other protocols in the presence and absence of adversaries. We simulate attacks on the routing metric, overhead packets, data packet, wormholes and rushing attacks, all aimed at either disrupting the route computation or giving the adversary control of data flows. We do not simulate fabrication, masquerading or replay attacks as these can easily be detected with cryptography.

We compare STOP to the Secure Routing through Diversity and Verification (SRDV) [7] protocol. SRDV is a recent routing protocol and is an amalgamation of several security paradigms. It constructs multiple paths based on a spatial ordering that is partially secured by cryptography and hash chains (much like SEAD [12]). It also uses load balancing based on end-to-end feedback like STOP. In short, SRDV incorporates some of the latest security features with the main differences being that STOP uses a time-based ordering whereas SRDV is based on spatial ordering, and that STOP uses all paths whereas SRDV only uses two paths. STOP is also compared to ARAN [23], which is based on AODV and designed for secure routing by stripping away the unsecured optimizations of AODV. We also include AODV as a baseline because it is one of the most well known routing protocol.

We also simulate SRDV-I with an idealized version of hash chains where nodes for somehow forced to hash the value to show the maximum theoretical performance. SRDV and SRDV-I would behave identically, except in the presence of adversaries which modify hop count.

The simulations were performed using the Qualnet 4.5 network simulator and the parameters are summarized in Table 2. This choice of parameters satisfies the minimum standards for rigorous MANET protocol evaluation as prescribed in [17], because it results in an *average shortest path hop count* [17] of 4.03 and *average network partitioning* [17] of 3.9 %. These parameters ensure that data packets travel several hops from source to the destination and thus test the robustness of the protocols. With this node density, each node has on average 7 neighbors with multiple paths to the destination. STOP was designed to take advantage of multiple paths, and the more paths there are the better it will be able to defend against attacks. If the network was set up so there there is only one or a few paths between the source and the destination, then STOP is behave more like the other protocols since there is little it can do to prevent or correct attacks on the data. We sampled the stationary distribution of node speed, remaining pause time and node placement according to the method outlined in [19] and used it as the initial conditions in the simulation. This ensured that the experiments started in steady state and provided consistent results.

7.1 Performance with no adversaries

Three metrics were used to evaluate and compare the performance of the protocols in the absence of adversaries. 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. The simulation results for the routing protocols tested are summarized in Table 3, where the mean and a 95 % confidence interval are given.

The effectiveness of STOP as a reliable routing protocol is evident from these results. STOP delivered almost 20 % more packets than SRDV while achieving the lowest

Table 2 Simulation parameters

Parameter	Value
Simulation time	900 s
Number of nodes	100
Simulation Area	1,000 m × 1,000 m
Node placement	Stationary
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
Radio range	150 m
MAC protocol	802.11 DCF
Data source	CBR
Number of packets per flow	400
Packet rate	4 packets per second
Number of flows	25
802.11 Data rate	2 Mbit/s

Table 3 Simulation results: no adversaries

	Delivery ratio	Latency	Net load
AODV	0.51 ± 0.7	0.09 ± 0.07	14 ± 3.3
DSR	0.18 ± 0.10	19.3 ± 12.4	5.0 ± 1.2
OLSR	0.35 ± 0.08	0.07 ± 0.03	68 ± 1.2
ARAN	0.40 ± 0.03	0.21 ± 0.09	22 ± 5.0
SRDV	0.64 ± 0.04	0.19 ± 0.07	3.9 ± 0.6
STOP	0.76 ± 0.07	0.07 ± 0.03	4.8 ± 1.2

average end-to-end delay. In terms of routing overhead, STOP incurred slightly more than SRDV but still significantly less than the other protocols. The main reason is the reduced frequency of route computations in STOP and SRDV as there are many available routes, but the periodic updates required to deliver the feedback is propagated by a larger number of nodes in STOP than SRDV. ARAN is based on AODV, but without many of the optimizations which were removed because of security vulnerabilities. Without these optimizations, ARAN's performance is notably worse than AODV and is a classic example of where security comes at the cost of performance.

7.2 Dropping data packets

Adversaries can sometimes gain access to data flows without taking any malicious action. If they happen to lie on the "best" path to the destination, they can perform the normal signaling and have data routed through them. There is no means to prevent this attack, especially if the adversary has no history of malicious behavior. If the adversary perform attacks by dropping the data packets or corrupting them, the attack can be discovered and countered using feedback and load balancing performed in STOP and SRDV. The detection will not be instantaneous and many packets can be lost before the attack is thwarted. Figure 5(a) shows the impact of this attack on the tested protocols. These results support the intuition discussed above (i.e., SRDV and STOP are less impacted than AODV and ARAN). STOP always routes a small fraction of data packets through nodes with very poor performance allowing them the opportunity to find better paths if they aren't adversaries. Consequently, this small fraction will always be lost to adversaries in the DAG.

7.3 Modification of hop count

One critical vulnerability of spatially ordered routing protocols is the manipulation of the distance metric. In particular, if an adversary advertises a distance to the destination that is shorter than its neighbors', it becomes the most desirable next hop. Spatially ordered routing protocols must therefore employ mechanism (such as the hash chains in SRDV and SEAD) to secure this field. Figure 5(b) shows the results when varying number of adversaries advertise distances that are smaller than their actual distance. These adversaries then drop any data packet routed through them. AODV is unsecured, and not surprisingly suffers from the greatest decrease in performance. Protocols such as ARAN and STOP are immune to attacks on the routing metric, since they do not use an explicitly stated routing metric. However, packets can still be routed through adversaries which are actually on the best path to the destination and these data packets will be dropped. ARAN has no defense against this, but STOP will detect such attacks in the feedback and then use load balancing to avoid the path containing a suspected adversary. Hash chains, as used in SRDV, Ariadne and SEAD, are vulnerable to nodes hashing more than once or forwarding the hash value without hashing it themselves. The results for SRDV-I show that unless the hash chain is somehow enforced, this technique does not counter the attack as adversaries can significantly degrade the performance with a difference 1-hop (i.e. not hashing). Immunity to this attack is innate to STOP without the complexity of cryptography.

7.4 Dropping of route replies

Adversaries can attempt to disrupt route discovery, and thereby perform denial of service attacks by dropping route replies. The impact of this attack on the tested protocols are shown in Fig. 5(c). Protocols which establish a single path, such as AODV and ARAN are most vulnerable to these attacks. If the adversary lies on the single discovered path, it can prevent discovery by dropping the RREP. This attack has a lesser impact on SRDV and STOP as they establish multiple paths.

7.5 Rushing attacks

We simulate a simple rushing attack where adversaries retransmit overhead immediately to appear on the best path and the result is shown in Fig. 6(a). AODV and ARAN are especially vulnerable to this attack, but their poor performance even without adversaries makes it less noticeable. STOP uses all paths so there is no merit in rushing attacks and the load balancing based on performance ensures that adversaries are avoided if they attack the data. Likewise, SRDV takes countermeasures if the rushing attack succeeds in giving the adversary access to the data flow.

7.6 Wormhole attacks

Colluding adversaries can tunnel packets to each other using links not available to other nodes in the network.

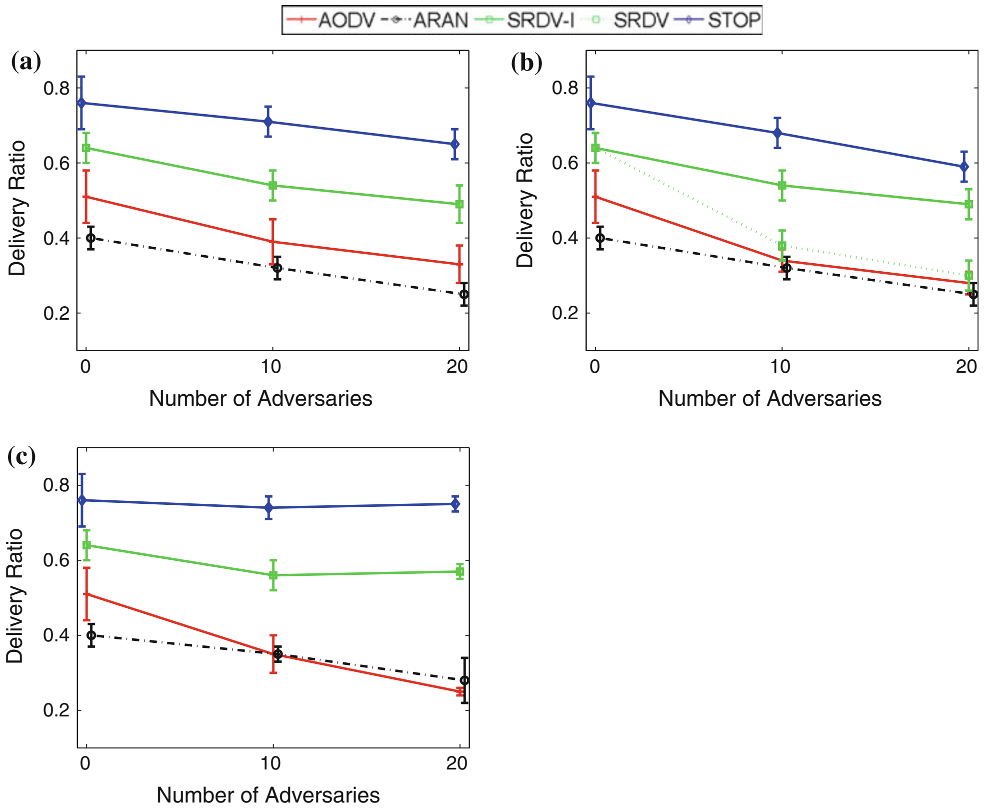

Fig. 5 Routing performance with various attacks. **a** Attacks on data packets. **b** Attacks on distance metrics. **c** Attacks on route replies

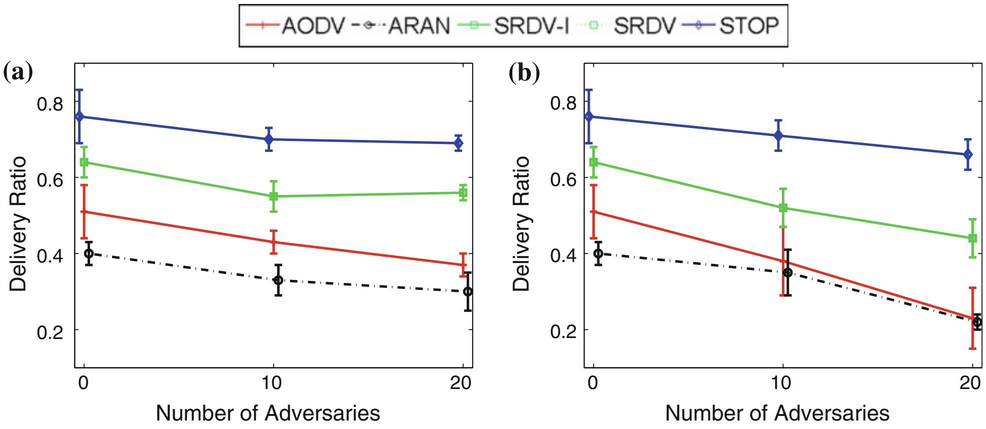

Fig. 6 Routing performance with various attacks. **a** Rushing attacks. **b** Wormhole attacks

This is called a wormhole attack [15] and these “wormholes” can also be used to perform rushing attacks. These nodes can make themselves seem closer to the destination without having to manipulate the metric field or time thus rendering countermeasures as the hash chain in SEAD useless.

In our simulations, of the 100 nodes in the network, we select five pairs randomly (ten distinct nodes) 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, nodes then drop all the data packets they receive. We repeated with 10 wormholes involving 20 distinct nodes and the results are given in Fig. 6(b). Any node connected to a tunnel is considered an adversary and drops all data packets.

This form of wormhole attacks cannot be detected without feedback from the destination, and once they are detected, choosing alternate paths is the only solution. By comparing these results to those with no adversaries, we can see that the simulated wormholes do present a threat, demonstrated by the reduced performance of AODV. However, these wormholes have a lesser impact on STOP and SRDV, which employ feedback with load balancing to detect and avoid these paths.

8 Conclusion

We have argued that previous solutions for securing routing in MANETs have significant limitations, and presented STOP as an instantiation of an approach based on ordering nodes in time rather than space. STOP implements on-demand routing and orders nodes based solely on the local time at which they receive and transmit signaling packets. This eliminates the ability of adversaries to manipulate distances, path or link information, which remains a significant vulnerability present in most on-demand and link-state protocols. The use of performance-driven path selection over an undistorted DAG serves to correct any attack which cannot be prevented in STOP. Our approach addresses many of the security problems previously identified with significantly less computational complexity.

References

1. Al-Shurman, M., Yoo, S. M., & Park, S. (2004). Black hole attack in mobile ad hoc networks. In *ACM-SE 42 Proceedings of the 42nd annual Southeast regional conference*.
2. Balachandran, R. K., Ramamurthy, B., Zou, X., & Vinodchandran, N. (2005). Crtdh: An efficient key agreement scheme for secure group communications in wireless ad hoc networks. In *IEEE ICC*.
3. Capkun, S., Buttyan, L., & Hubaux, J. P. (2003). Sector: Secure tracking of node encounters in multi-hop wireless networks. In *ACM workshop on security of ad hoc and sensor networks*.
4. Choi, S., Kim, D. Y., Lee, D. H., & Jung, J. I. (2008). Wap: Wormhole attack prevention algorithm in mobile ad hoc networks. In *IEEE international conference on sensor networks, ubiquitous and trustworthy computing*.
5. Cordasco, J., & Wetzel, S. (2008). Cryptographic vs. trust-based methods for manet routing security. *Electronic Notes in Theoretical Computer Science*, *197*(2), 131–140.
6. Dabideen, S., & Garcia-Luna-Aceves, J. (2010). Ordering in time: A new routing approach for wireless networks. In *Proceedings of IEEE MASS*.
7. Dabideen, S., Smith, B. R., & Garcia-Luna-Aceves, J. J. (2010). An end-to-end approach to secure routing in manets. *Wiley's Security and Communication Networks Journal; Special Issue on Security in Mobile Wireless Networks, 3*(2–3), 130–149.
8. Deng, H., Li, W., & Agrawal, D. P. (2002). Routing security in ad hoc networks. *IEEE Communications Magazine. Special Topics on Security in Telecommunications Networks, 40*(10), 70–75.
9. Dutta, R., Mukhopadhyay, S., & Collier, M. (2010). Computationally secure self-healing key distribution with revocation in wireless ad hoc networks. *Ad Hoc Networks, 8*(6), 597–613.
10. Ericksson, J., Faloutsos, M., & Krishnamurthy, S. V. (2007). Routing amid colluding attackers. In *Proceedings of ICNP*.
11. Ewy, B. J., Swink, M. T., Pennington, S. G., Evans, J., Kim, J. M., Ling, C., et al. (2009). Tigr in iraq and afghanistan: Network-adaptive distribution of media rich tactical data. In *IEEE military communications conference*.
12. Hu, Y. C., Johnson, D. B., & Perrig, A. (2003). Sead: Secure and efficient distance vector routing for mobile wireless ad hoc networks. *Ad Hoc Networks*, *1*, 175–192.
13. Hu, Y. C., Perrig, A., & Johnson, D. B. (2003). Packet leashes: A defense against wormhole attacks in wireless networks. In *Proceedings of INFOCOM*.
14. Hu, Y. C., Perrig, A., & Johnson, D. B. (2005). Ariadne: A secure on-demand routing protocol for ad hoc networks. *Wireless Networks*, *11*, 21–38.
15. Hu, Y. C., Perrig, A., & Johnson, D. B. (2006). Wormhole attacks in wireless networks. *IEEE Journal on Selected Areas in Communications*, *24*(2), 370–380.
16. Johnson D. B., Maltz D. A., & Broch J. (2001). *DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks*. Reading, MA: Addison-Wesley.
17. Kurkowski, S., Camp, T., & Navidi, W. (2006). *Minimal standards for rigorous MANET routing protocol evaluation*. Technical Report MCS 06-02, Colorado School of Mines.
18. Marti, S., Giuli, T., Lai, K., & Baker, M. (2000). Mitigating routing misbehavior in mobile ad hoc networks. In *Proceedings of ACM MOBICOM*.
19. Navidi, W., & Camp, T. (2006). Stationary distributions for the random waypoint mobility model. *IEEE Transactions on Mobile Computing*, *3*(1), 1153–1166.
20. Pathak, V., Yao, D., & Iftode, L. (2008). *Securing geographical routing in mobile ad-hoc networks*. Rutgers University, Technical Report 638.
21. Perkins, C. E., & Bhagwat, P. (1994). Highly dynamic destination-sequenced distance-vector routing (dsdv) for mobile computers. In *Proceedings of SIGCOMM'94* (pp. 234–244).
22. Ramaswamy, S., Fu, H., Sreekantaradhya, M., Dixon, J., & Nygard, K. (2003). Prevention of cooperative black hole attack in wireless ad hoc networks. In: *Proceedings of international conference on wireless networks*.
23. Sanzgiri, K., Dahill, B., Levine, B. N., Shields, C., & Belding-Beyer, E. (2002) A secure routing protocol for ad hoc networks. In *Proceedings of ICNP*.
24. Sivakumar, K. A., & Ramkumar, M. (2007). An efficient secure route discovery protocol for DSR. In *IEEE GLOBECOM*.
25. Xiaopeng, G., & Wei, C. (2007). A novel gray hole attack detection scheme for mobile ad-hoc networks. In *IFIP international conference on network and parallel computing workshops*.
26. Yi, S., Naldurg, P., & Kravets, R. (2001). Security-aware ad hoc routing for wireless networks. In *Proceedings of ACM Mobihoc*.
27. Hu, Y.-C., & Adrian Perrig, D. B. J. (2003). Rushing attacks and defense in wireless ad hoc network and routing protocols. In *Proceedings of WiSe*.

Author Biographies

Step**hen Dabideen** received his B.S. degree in Computer and Telecommunications Engineering at the University of Pennsylvania in 2006. He earned a Ph.D. in Computer Engineering at the University of California, Santa Cruz where he focused on networking in Computer Networks under the supervision of Prof. J.J. Garcia-Luna-Aceves. He is currently a Network Scientist at Raytheon BBN Technologies in Cambridge, MA. His research interests are wireless networks, delay tolerant networking and the applications of machine learning techniques to networks.

J. J. Garcia-Luna-Aceves received the B.S. degree in Electrical Engineering from the Universidad Iberoamericana, Mexico City, Mexico in 1977; and the M.S. and Ph.D. degrees in Electrical Engineering from the University of Hawaii at Manoa, Honolulu, HI in 1980 and 1983, respectively. He holds the Jack Baskin Endowed Chair of Computer Engineering at the University of California, Santa Cruz (UCSC), is Chair of the Computer Engineering Department, and is a Principal Scientist at the Palo Alto Research Center (PARC). Prior to joining UCSC in 1993, he was a Center Director at SRI International (SRI) in Menlo Park, California. He has been a Visiting Professor at Sun Laboratories and a Principal of Protocol Design at Nokia. Dr. Garcia-Luna-Aceves holds 35 US patents, and has published three books and more than 400 journal and conference papers. He has directed 30 Ph.D. theses and 28 M.S. theses since he joined UCSC in 1993. He has been the General Chair of the ACM MobiCom 2008 Conference; the General Chair of the IEEE SECON 2005 Conference; Program Co-Chair of ACM MobiHoc 2002 and ACM MobiCom 2000; Chair of the ACM SIG Multimedia; General Chair of ACM Multimedia'93 and ACM SIGCOMM'88; and Program Chair of IEEE MULTIMEDIA'92, ACM SIGCOMM'87, and ACM SIGCOMM'86. He has served in the IEEE Internet Technology Award Committee, the IEEE Richard W. Hamming Medal Committee, and the National Research Council Panel on Digitization and Communications Science of the Army Research Laboratory Technical Assessment Board. He is an IEEE Fellow and an ACM Fellow, and is listed in Marquis Who's Who in America and Who's Who in The World. He is the co-recipient of the IEEE Fred W. Ellersick 2008 MILCOM Award for best unclassified paper. He is also co-recipient of Best Paper Awards at the European Wireless Conference 2010, IEEE MASS 2008, SPECTS 2007, IFIP Networking 2007, and IEEE MASS 2005 conferences, and of the Best Student Paper Award of the 1998 IEEE International Conference on Systems, Man, and Cybernetics. He received the SRI International Exceptional-Achievement Award in 1985 and 1989.