Performance Analysis of Secure Multipath Routing Protocols for Mobile Ad Hoc Networks

Rosa Mavropodi, Panayiotis Kotzanikolaou, Christos Douligeris

University of Piraeus
Department of Informatics
80 Karaoli & Dimitriou, Piraeus 185 34, Greece
{rosa,pkotzani,cdoulig}@unipi.gr

Abstract. In the Mobile Ad Hoc Network (MANET) paradigm, multipath routing protocols were initially proposed due to QoS needs, since they do not require initiation of the route discovery process after each link disconnection. Moreover, research on MANET routing security has shown that multipath routing provides increased resilience against security attacks of collaborating malicious nodes. Towards this direction, several secure multipath routing protocols have been recently proposed in the literature, which indeed provide such increased security protection for critical applications. However, embedding security mechanisms always imposes extra burden to the route discovery process. In this paper, we evaluate the performance of the existing secure multipath routing protocols for MANET through extensive simulations in various traffic scenarios.

1 Introduction

Multipath routing protocols were initially proposed to ensure QoS in mobile ad hoc networks. Maintenance of multiple routes towards a destination, prevents initiation of a new path discovery from the source node, each time there is a link failure. Furthermore, the existance of multiple paths may prevent node congestion, since it balances the traffic load through alternative routes. Examples of ad hoc multipath routing protocols are given in [9, 7, 10, 11, 5]. The route discovery may stop when a sufficient number of paths is discovered (*e.g.* [11]) or when all possible paths are detected (*e.g.* [3]). The protocols of the second case, are also known as complete. Multipath routing protocols can be *node-disjoint* (*e.g.* [11]) or *link-disjoint* (*e.g.* [6]) if a node (or a link) cannot participate in more than one path between two end nodes.

Apart from the multipath routing protocols that aim to increase efficiency, several multipath routing protocols have been proposed,recently, in order to provide additional security services. More specifically, the secure multipath routing protocols of [8, 2, 4] were designed in order to resist Denial of Service (DoS) attacks of collaborating malicious nodes, which single path protocols fail to encounter. Indeed, with single path routing protocols it is trivial for an adversary to launch a DoS attack, even if security measures are taken. A malicious node

controlled by the adversary may participate passively in the routing path between two end nodes and may behave as a legitimate intermediate node. The malicious node can stop the communication at any time it seems most advantageous to the adversary. Although communication may be cryptographically protected, network characteristics (such as variation in traffic) or external factors may be used by the adversary in order to identify the proper time to disrupt communication. Even though the end nodes may initiate a new route request after the DoS attack, the time required to establish the new path may be critical. A dedicated and skilful adversary may thus identify the most critical nodes and disable their single routing paths, by compromising a small fraction of nodes.

Multipath routing protocols can be resilient to DoS attacks and may protect network availability from faulty or malicious nodes [1]. Indeed, if there exist k node-disjoint paths between two end nodes, the adversary should compromise at least k nodes - and more particularly at least one node in each path - in order to control their communication. A secure multipath routing protocol must be node-disjoint. Otherwise, a malicious node would be allowed to participate and consequently control more than one path. Thus, a single malicious node may manipulate the routing protocol and in this way it may compromise all the available routes between two end nodes.

In order to achieve resilience to DoS attacks, a multipath routing protocol should be properly enhanced with cryptographic means, which will guarantee the integrity of a routing path and the authenticity of the participating nodes. Towards this direction, three secure multipath routing protocols have been recently proposed; the Secure Routing Protocol (SRP) [8], the multipath routing protocol of [2] and the Secure Multipath Routing protocol (SecMR) [4]. The secure multipath routing protocols of [8, 2, 4] may guarantee at a certain level the availability of the communication against DoS attacks of a bounded number k of collaborating malicious nodes, by employing $k+1$ node-disjoint routing paths between two communicating nodes. However, the cryptographic protection in the route discovery of the secure multipath routing protocols will naturally increase the control overhead and until now, the efficiency of the secure multipath routing protocols for ad hoc networks has not been estimated.

In this paper, we evaluate the performance of the currently proposed secure multipath routing protocols of [8, 2, 4] by simulating their behavior in various traffic scenarios. In section 2, we briefly describe the examined routing protocols. In section 3, we present the simulation results, while in section 4 we discuss possible enhancements and we conclude this paper.

2 A description of the examined secure multipath routing protocol

In this section we briefly describe the route discovery process of the examined secure multipath routing protocols, along with some comments on the security properties of each protocol.

2.1 The SRP protocol

SRP [8] was initially developed having in mind general security considerations of ad hoc networks. The basic considerations of the SRP protocol are integrity protection of the routing paths and authentication of the end nodes. The route discovery of the SRP can be used to discover multiple node-disjoint paths.

Before the propagation of a route request query, the source node assigns to it unique identifiers, in order to avoid replay attacks. When an intermediate node receives a route request, it checks whether it has already processed a query originating from the particular source node with the same identifiers, in order to drop it. Otherwise, it adds itself to the routing path and it forwards the request. In this way, an intermediate node can only participate in a single path between two end nodes and the paths that will be discovered will be node-disjoint. When the target node receives a route request query, the node checks the authenticity of the request be using a symmetric encryption key - a security association - which the two end nodes are supposed to share prior to the request. The route reply query will also be protected with the same security association, in order to protect the integrity of the routing paths. The protocol finds a number of node-disjoint routing paths between the source and the destination, which can be used for multipath communication.

The SRP protocol is very efficient since it restricts security checks at the end nodes only and it uses efficient symmetric key encryption. A problem of this protocol is that it does not authenticate the intermediate nodes which may lead to several impersonation attacks and in this way reduce the resilience of the protocol to DoS attacks [4]. For example, a malicious intermediate node may participate with fake identities to several paths, rendering multipath routing insecure. Furthermore, the protocol is not complete in the discovery of the existing node-disjoint multiple paths, *i.e.* although the paths discovered are node-disjoint the protocol may not discover all the existing node-disjoint paths between the two end nodes, depending on the propagation conditions of the query.

2.2 The secure multipath routing protocol of Burmester and Van Le

The secure multipath routing protocol of [2] is based on the Ford-Fulkerson MaxFlow algorithm. The propagation of a route request query assures that a query will reach any intermediate node within a pre-defined maximum hop distance. During the route request propagation, a node that receives a route query message for the first time, appends its neighborhood information along with a signature and re-broadcasts the message along with all the previously received query information. When the request query message reaches the destination, the destination node uses the received information in order to estimate the current connectivity of the intermediate nodes that answered the request query. In this way, the destination node can construct the complete set of the existing node-disjoint paths.

This protocol satisfies all the security requirements of multipath routing, since it authenticates all participating nodes, while it also protects the integrity of the routing paths. Furthermore, it satisfies completeness, *i.e.* it discovers all existing paths bounded with a TTL or maximum hop field. However, the propagation of the route request query is not efficient in terms of computation and space costs. The message size of a route request may increase to intolerable levels, since it contains information regarding the connectivity of all previous nodes. Furthermore, the use of digital signatures by the intermediate nodes of each route request message costs both in delay and processing power and may not be affordable for typical equipment.

2.3 The SecMR protocol

In order to reduce the cost of node authentication, the SecMR [4] protocol works in two phases. The first phase is the neighboring authentication phase which is repeated in periodic time intervals. During this phase, nodes in range are mutually authenticated through digital signatures. Each node n_i constructs a set N_i that contains the identifiers of its authenticated neighbors. The neighborhood set is then used in the second phase of the protocol, which involves the route discovery. The advantage of using a separate authentication phase is that the number of signatures and verifications performed by each node is bounded in each authentication period and does not depend on the number of paths that the node will participate in for a given authentication period.

A route request message in the SecMR protocol contains three independent lists of nodes, in order to reduce the cost of the route discovery. The *RouteList* is the list of the intermediate nodes participating in a routing path. The $NexHop$ list contains the possible next participants of a particular route query. Finally, the *ExcludeList* holds the nodes that are not allowed to participate in the particular instance of the route request query.

An intermediate node n_i receiving a query will process the query, provided that: i) it is listed in the $NextHop$ list, ii) it does not already belong to the *RouteList* and iii) it is not listed in the *ExcludeList*. Processing the request involves updating the lists included in the query. The updated *RouteList*, is constructed by appending its identifier to the received *RouteList*. The updated *ExcludeList* is generated by appending the rest of the nodes included in the received $NextHop$ list, into the received *ExcludeList* (duplicates are removed). Finally, the updated $NextHop$ list is generated as the list of the neighbors N_i of the node n_i that executes the route query (again, node identifiers already participating in another list are removed). Now, the node n_i updates the query thread with the new lists and broadcasts it.

The use of the *ExcludeList* and the $NextHop$ list is a key element for the efficient propagation of a route request. The $NextHop$ list restricts the query to propagate only through mutually authenticated nodes. The use of the *ExcludeList* dynamically generates non-cyclic "threads" of the request in an optimized way. By dynamically generating threads of a request, the algorithm

eventually discovers all the existing node-disjoint paths for a pre-defined maximum hop distance and only a limited number of redundant paths. To clarify the threading of a request query, consider the following scenario. Let n_i be an intermediate node that broadcasts the request $Q_{S,T}$ for the source and target nodes S and T respectively, to its neighbors n_j, n_k, after it has processed it (see figure 1). In order to distinguish the various threads of the request query, we denote as $Q_{S,T/i}$ the thread that is processed by node n_i. Thus, the thread $Q_{S,T/i}$ will contain the lists: $RouteList = \{X, ID_i\}$, $ExcludeList = \{Y\}$ and $NextHop = \{ID_j, ID_k\}$, where X and Y denote sequences of node identifiers.

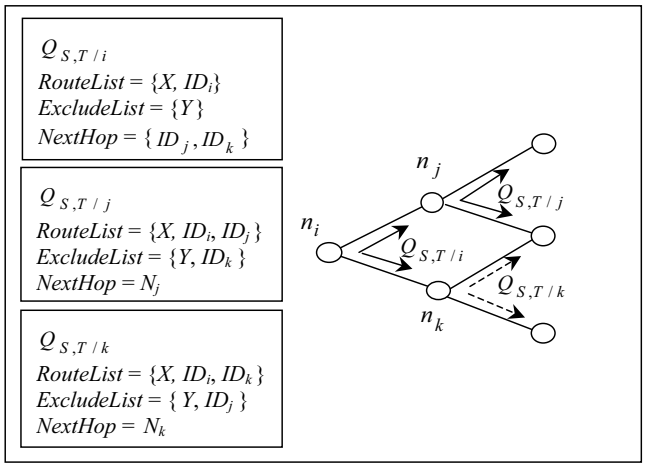

Fig. 1. Threading of a route request query

Both nodes n_j and n_k will process the request (supposing that $ID_j, ID_k \notin X, Y$). Node n_j will add its identifier to the $RouteList$, add the identifier of n_k to the $ExcludeList$ and update the $NextHop$ list with its own neighborhood. Thus, the updated threads of the request query will become $Q_{S,T/j}$, $Q_{S,T/k}$, containing the updated lists shown in figure 1. Each of these threads will propagate towards T, with the limitation that the thread $Q_{S,T/j}$ is not allowed to pass from the node n_k and vice versa. This forces the query to move only to more distant nodes of S towards T. The threads that return backwards tend to decline in a short time, when they reach a node closer to S that has been previously excluded.

At the end of the route discovery, the target and the source nodes will use a symmetric key contained in the route request message, in order to verify the integrity of the discovered paths.

3 Efficiency Analysis

Our study involves a comparison of the route request query between the SRP protocol [8], the complete multipath routing protocol of [2] and the SecMR pro-

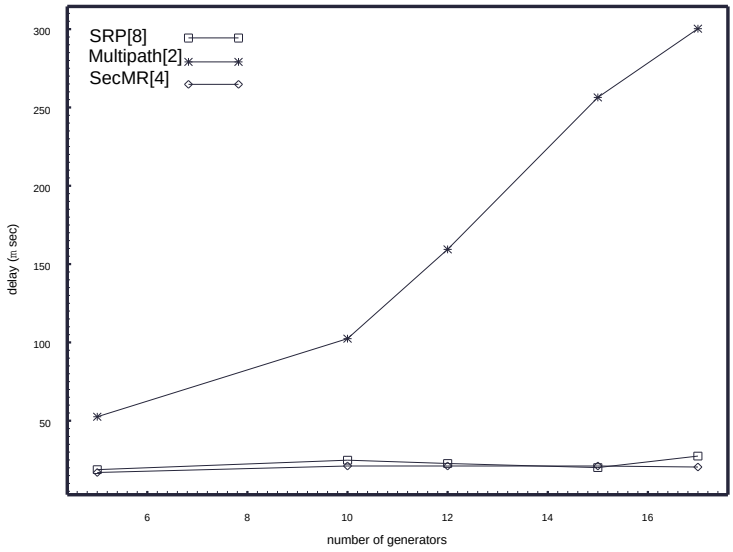

Fig. 2. Total data packet delay for send rate 1 pkt/sec

tocol [4]. We implemented the simulator within the NS-2 library. Our simulation modelled a network of 50 hosts placed randomly within a $670 \times 670m^2$ area. Each node has approximately 5 hops as neighbors. Each node has a radio propagation range of 150 meters and channel capacity was 2 Mb/s. The minimum and maximum speed is set to 0 and 20 m/s, respectively. This setup leads to a relatively dense network distribution with medium to high mobility and with medium mean connectivity. The size of the data payload was 512. Each run executed for 600 sec of simulation time. We used the IEEE 802.11 Distributed Coordination Function (DCF) as the medium access control protocol. The traffic generators were developed to simulate constant bit rate sources. The sources and the destinations are randomly selected with uniform probabilities. The destination of the traffic wait for 5 seconds until it assumes that all possible paths have been found. We generated various traffic scenarios by using different number of sources and scalar data send rate.

A free space propagation model with a threshold cutoff was used in our experiments. In the radio model, we assumed the ability of a radio to lock onto a sufficiently strong signal in the presence of interfering signals, *i.e.*, radio capture.

Figure 2 shows the average delay of the received data packets for a send rate equal to 1 pkt/sec. We can observe from the results that both SRP [8] and SecMR [4] outperform the multipath protocol of [2] especially when the number of data generators increases, which depicts high traffic conditions. In both SRP and SecMR the number of generated messages during the route discovery process are kept in sufficient low levels while the ones of Multipath[2] tent to flood the network. This is because in the multipath protocol of [2], each intermediate node

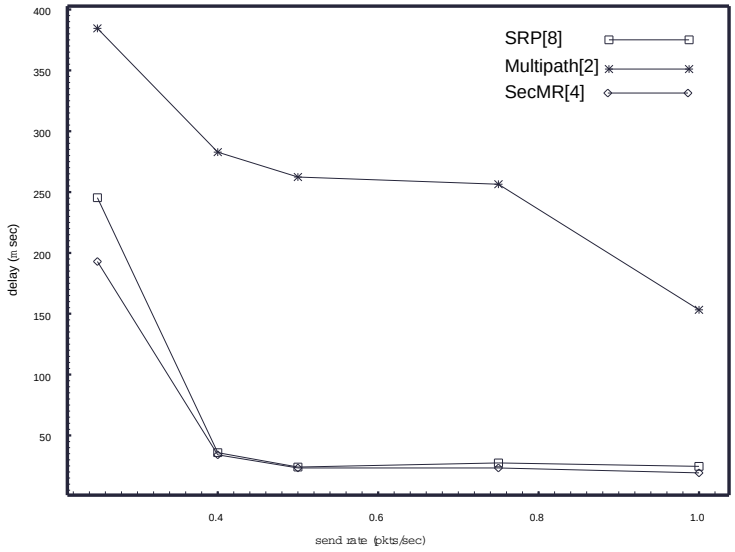

Fig. 3. Total data packet delay for 17 data sources

forwards all the route requests that reaches it for a given source, destination and sequence number, while SRP forwards only the first and SecMR performs a selective forward with the use of the exclude list. This flooding of the network results in higher delay in the data packet delivery. On the other hand, as the node's movement is rather high, the discovered paths of SRP are insufficient thus resulting in degradation of its performance. Figure 3, which presents the average delay of the received data packets to a network with multiple data sources with scalar send rate, strengthens the above observations. Indeed, as shown in figure 3, the SecMR protocol handles high traffic conditions better.

Figures 4 and 5 present the average total time that route request messages travel through the network. Figure 4 presents the average total time that route request messages travel through the network versus the number of data generators, for a send rate of 1 pkt/sec. Again, an increase in traffic leads to a proportional increase of the time that the route request messages are alive. In the secure multipath protocol of [2] a route request travels for a longer time than in the other two protocols, as the request is being forwarded to all nodes in range, many of which will not be included into one of the discovered paths. The route request of the SRP propagetes the request towards the destination faster than the other protocols, since it rejects any variant of a specific request. The route request of the SecMR has slightly longer living time than SRP. This is reasonable as it attempts to ensure descovery of the complete set of existing node-disjoint paths. Furthermore, the SecMR makes sure that all its neighboring nodes have contributed to the route discovery, either by participating to the *RouteList* (*i.e.* to a routing path) or by avoiding to re-process the same thread

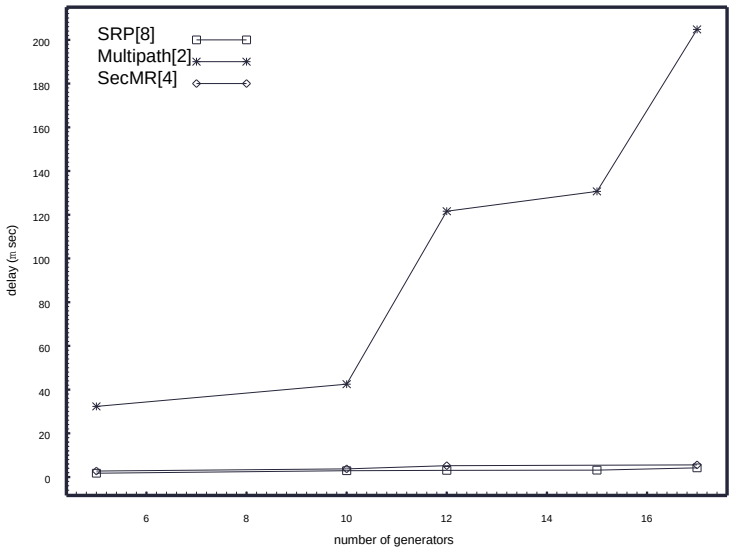

Fig. 4. Average living time of route request messages for send rate 1 pkt/sec

of the query (*i.e.* by participating into the *ExcludeList* of the query thread). This is also obvious in figure 5, which presents the average total time that a request travels through the network, per send rate, for multiple data sources.

Figures 6 and 7 illustrate the average time taken for a request message to reach the destination. In dense traffic conditions, while in SRP and SecMR the required time is comparable to the average time the request stays alive in the network (as illustrated in figures 4 and 5 respectively), in the case of the protocol of [2] the request stays alive in the network almost 8 times more after the first request query thread has reached its destination. This means that the redundant request messages will exist in the network for a long time, causing the network to experience high delays. Finally, figures 8 and 9 illustrate the average throughput, per send rate, of data and route control messages respectively for 17 data sources. Multipath serves less data packets then SecMR and SRP (figure 8) in contrast to the number of the routing control messages (figure 9).

4 Discussion and Future Work

The simulation results provide significant evidence about the efficiency of the route request propagation of the examined secure multipath routing protocols. However, it should be mentioned that these ratings reflect the examined network design scenario, with relatively dense network distribution, medium to high mobility and medium mean node connectivity. In this scenario, SRP performs better than the other two protocols, SecMR follows in short distance, while the protocol of [2] seems to be the heavier.

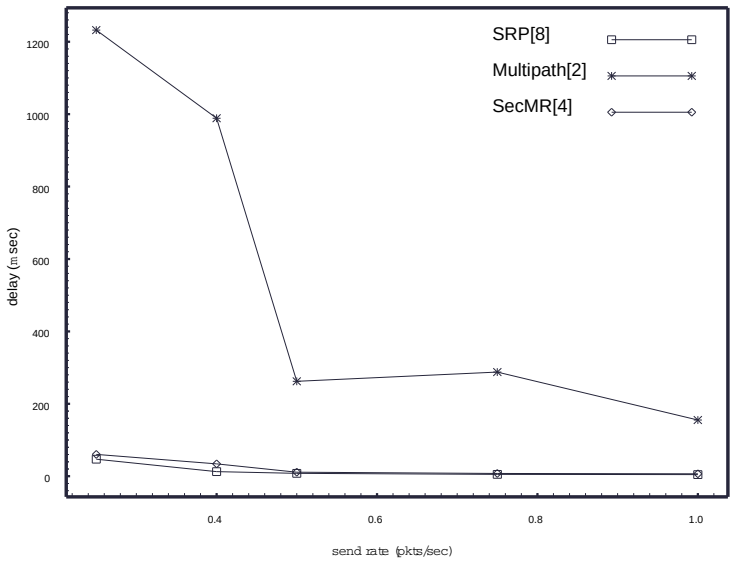

Fig. 5. Average living time of route request messages for 17 data sources

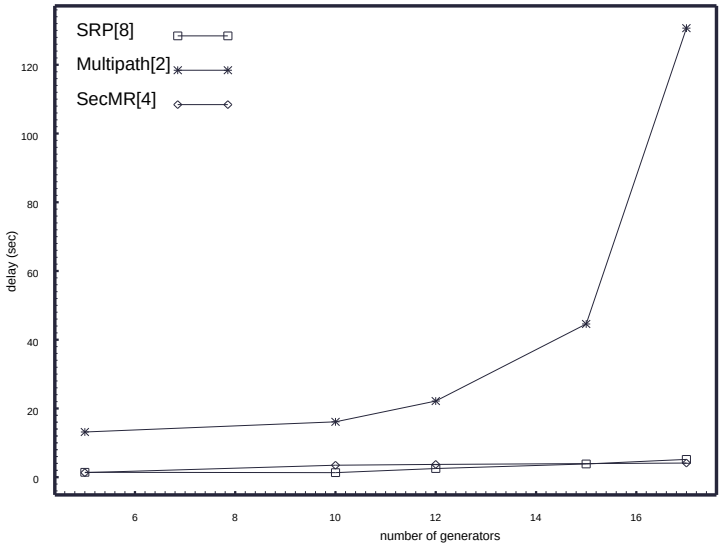

Fig. 6. Route Discovery delay for send rate 1 pkt/sec

From a security point of view, the ranking is reversed. The protocol of [2] achieves all the required security properties, to provide maximum resilience against DoS attacks of collaborating malicious nodes. It provides completeness in the route discovery process and it explicitly authenticates all the intermediate

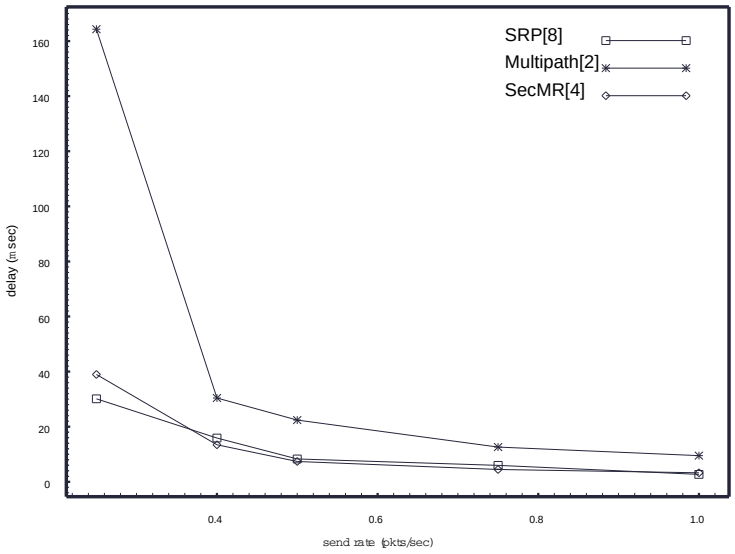

Fig. 7. Route Discovery delay for 17 data sources

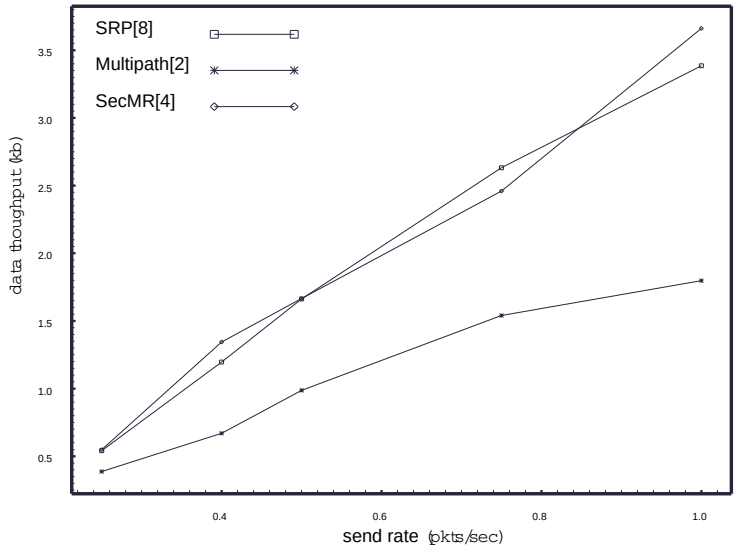

Fig. 8. Average throughput of data packets for 17 data sources

nodes in each routing path. The SecMR protocol also achieves completeness and it provides implicit authentication of the intermediate nodes, since node authentication is performed once for a discrete time period. Finally, the SRP protocol

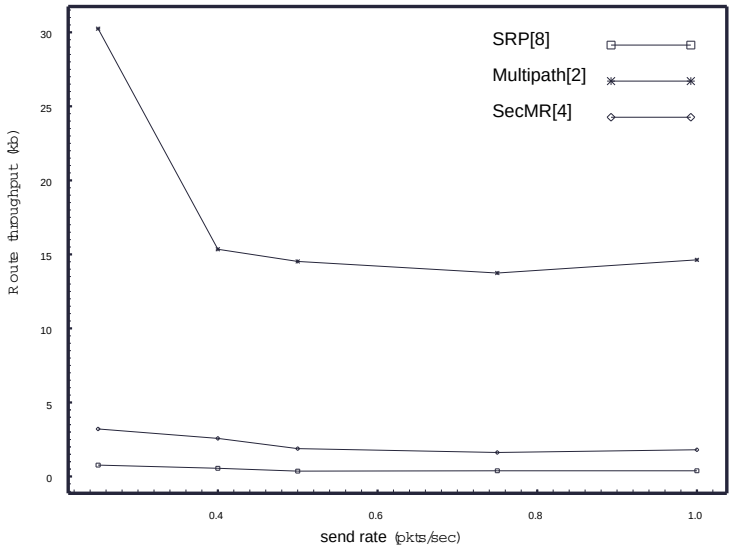

Fig. 9. Average throughput of route control packets

does not provide the complete set of node-disjoint paths, and it provides only end-to-end authentication.

Based on the above observations and the simulation results, the protocol of [2] can be considered suitable for security critical ad hoc network applications, but its applicability can only be considered in networks with low mobility and relatively low density. In situations with high mobility and high node density, the protocol would saturate the network, since it would lead to long route request messages which would exist in the network for long time.

The SecMR protocol seems most appropriate for ad hoc networks that require high security protection and they present medium to high mobility and medium node density. Indeed, in such situations the SecMR protocol has comparable efficiency with the SRP, while it offers increased security level. Moreover, as the node mobility increases, the SecMR shows better performance than the SRP. This is due to the fact that the SRP discovers less paths than the other two protocols and this forces the protocol to re-initiate route requests in shorter time that the other protocols, when nodes move and links are broken.

Finally, the SRP seems a suitable choice for several network configurations with increased node density. This is caused by the fact that the route request propagation avoids discovery of all the possible routes that each node could participate and in this way it converges faster. This however leads to a non-complete route discovery [4] and reduces the security resilience of the protocol to distributed DoS attacks. Thus, the SRP seems suitable for applications with medium security risks.

Regarding possible extensions of our work, we consider examining the behavior of the secure multipath routing protocols in various network configurations and arrival patterns. Furthermore, we consider examination of the behavior of the route reply and route maintenance algorithms of the examined protocols.

References

1. M. Burmester and Y. Desmedt, *Secure communication in an unknown network using certificates*, Advances in Cryptology - Asiacrypt '99, Lecture Notes in Computer Science Vol. 1716, Springer, 1999, pp. 274–287.
2. M. Burmester and T. van Le, *Secure multipath communication in mobile ad hoc networks*, Proceedings of the International Conference on Information Technology: Coding and Computing (ITCC 2004) (Las Vegas), IEEE, April 2004.
3. Gunyoung Koh, Duyoung Oh, and Heekyoung Woo, *A graph-based approach to compute multiple paths in mobile ad hoc networks*, Lecture Notes in Computer Science Vol.2713, Springer, 2003, pp. 3201–3205.
4. P. Kotzanikolaou, R. Mavropodi, and C. Douligeris, *Secure multipath routing for mobile ad hoc networks*, Proceedings of the WONSS'05 Conference (St. Moritz, Switzerland), IEEE, January 2005.
5. Sung-Ju Lee and Mario Gerla, *Split multipath routing with maximally disjoint paths in ad hoc networks*, Proceedings of ICC 2001 (Helsinki, Finland), IEEE, June 2001, pp. 3201–3205.
6. Mahesh K. Marina and Samir R. Das, *Ad hoc on-demand multipath distance vector routing*, ACM SIGMOBILE Mobile Computing and Communications Review **6** (2002), no. 3.
7. A. Nasipuri and S.R. Das, *On-demand multipath routing for mobile ad hoc networks*, Proceedings of IEEE INFOCOM99, 1999, pp. 64–70.
8. P. Papadimitratos and Z. Haas, *Secure routing for mobile ad hoc networks*, In Proceedings of the SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS) (TX, San Antonio), January 2002.
9. Anand Prabhu Subramanian, A. J. Anto, Janani Vasudevan, and P. Narayanasamy, *Multipath power sensitive routing protocol for mobile ad hoc networks*, Lecture Notes in Computer Science Vol.2928, Springer, 2003, pp. 171–183.
10. A. Tsirigos and Z.J. Haas, *Multipath routing in the presence of frequent topological changes*, IEEE Communications Magazine **39** (2001), no. 11, 132–138.
11. Jie Wu, *An extended dynamic source routing scheme in ad hoc wireless networks*, Telecommunication Systems **22** (2003), no. 1-4, 61–75.