

Robust Routing in Malicious Environment for Ad Hoc Networks [★]

Zhongchao Yu¹, Chuk-Yang Seng¹, Tao Jiang², Xue Wu¹, and
William A. Arbaugh^{1,3}

¹ Dept. of Computer Science, University of Maryland, College Park, MD 20742, USA
{yuzc, sengcy, wu, waa}@cs.umd.edu

² Dept. of Electronic and Engineering, University of Maryland, College Park,
MD 20742, USA
{tjiang}@glue.umd.edu

³ University of Maryland Institute of Advanced Computer Studies, College Park,
MD 20742, USA

Abstract. Secure routing in ad hoc networks has been extensively studied in recent years. The vast majority of this work, however, has only focused on providing authenticity of the route. Availability of the network in a malicious environment has largely been ignored.

In this paper, we divide the secure routing problem into two layers. The first layer provides authenticated routing and the second layer provides a route selection algorithm that selects a route with the highest probability of successful delivery rather than the shortest route. We provide a metric for evaluating this probability. We provide simulation results that demonstrate that our approach increases the throughput by at least ten percent in a network where fifty percent of the nodes are malicious when compared to an approach that selects the shortest route. Furthermore, our approach incurs only a small delay when compared to the delay along the shortest route.

1 Introduction

A mobile ad hoc network (MANET) is a dynamic collection of wireless nodes, communicating with each other over possible multi-hop paths. There is no pre-defined infrastructure to provide services, and each node within the network acts as a router, forwarding data packets for other nodes. Because each node acts as a router, the correct behavior of each node is vital to the efficacy of the network. In an adversarial environment, we always wish to make sure that our data packet falls into trusted hands.

Previous works on secure routing protocol focused on providing authenticated routes, by which we mean that each node on the route is authenticated. We call these protocols *authenticated routing protocols* in this paper. Some examples include Ariadne [1], SAODV [4], ARUN [5] and etc. Authenticated routing

[★] Portions of this work were funded by a Critical Infrastructure Grant from the U.S. National Institute of Standards and Technology

protocols prevent many of the attacks such as message fabrication, packet modification, impersonation and so on. However, as detailed in Section 2, authenticated routing protocols alone do not guarantee the correct behavior of the nodes. This is because authentication only verifies identity. It does not guarantee that the node will behave correctly. Therefore, these protocols must be augmented by other approaches which select routes on which nodes are not only authenticated, but also exhibit correct behavior in forwarding data.

We introduce a technique based on risk analysis to improve the robustness of routing. By robustness of routing, we are referring to the likelihood of successful delivery of data packets. To improve the robustness of routing in a malicious environment, we seek to choose the route that provides the minimum risk. We evaluate risk based on (partial) trust relationships between nodes.

We build a protocol based on risk – Risk-Based Protocol (RBP) to complement authenticated routing protocols. A protocol intended to secure routing in ad hoc networks without authenticating the nodes and routes is bound to be vulnerable to Sybil attacks [8]. While a protocol that ignores the behavior of the nodes within the network will suffer greater packet loss as the number of malicious nodes increases. An important insight is recognizing that *both* approaches must be used to provide secure routing.

Consequently, we present a novel routing paradigm for securing ad hoc networks. Our routing paradigm consists of a two layer architecture. The bottom layer provides authenticated routes. While the top layer evaluates risk by estimating the behavior of nodes on the route. This is the first paper which explicitly separates behavior from authentication in securing ad hoc routing protocols as well as recognizing the importance of using both together. The work in this paper, RBP, focuses on the top layer of the architecture– evaluation and selection of routes.

The rest of the paper is organized as follows. We will detail the rationale of this work in Section 2, followed by related work in Section 3. Then we introduce the trust model and assumptions we used in our work in Section 4, followed by the actual protocol in Section 5 and the evaluation of the protocol in Section 6. We conclude in Section 7.

We will use the following notation in the remaining parts of the paper.

1. We always use S to represent the source node and D to represent the destination node.
2. We use R to refer to a specific route.

2 Motivation

In networking, we need to make sure that any ID claimed by a node really exists. Otherwise a node can claim to be any other node. However, even if an ID is authentic, it does not mean that it represents a well-behaved node. For example, in a military scenario, nodes can be captured by the enemy and their key material for authentication may be disclosed. Moreover, it may be possible for intruders to exploit any software vulnerabilities to take control of a node. In

such situation, the compromised node may carry out malicious acts without the knowledge of the node owner.

In the context of routing, an authenticated routing protocol thwarts attacks such as routing message modification, impersonation and packet fabrication. However, it is still possible for authenticated nodes to behave maliciously. It is quite possible for intermediate nodes to follow the protocol faithfully in the route discovery phase and then drop data packets later.

To handle the deficiencies of authenticated routing protocols, we must consider the behavioral aspect of the nodes along a route. In this paper, we use partial trust information to predict the behavior of nodes. We will define our notion of trust later.

Ideally, whenever a node S sends a data packet, if it can always find a route on which every node is trusted by S to behave appropriately, then we are done. If no such route can be found, the only thing we can do is to find a route with a minimum probability of being compromised. This process is all about risk inherently.

3 Related Work

Much of the previous works on securing routing protocols for ad hoc networks have focused on authenticating mobile nodes and routing messages [1, 2, 4, 5], while a smaller amount researches have focused on the behavior aspects of the nodes within the network [6, 7, 13].

For authenticated routing protocols, either the authenticity of nodes on the source route or the authenticity of metrics critical to the selection of routes, such as number of hops, is guaranteed. Ariadne [1], ARUN [5] and BISS [13] are examples of protocol that provides authenticity of nodes. As previously argued, authentication alone is insufficient.

SEAD [2] and SAODV [4] are two protocols which authenticate the number of hops or sequence numbers in the packets using hash chains. It does not assure integrity of the source or destination. Thus they have to rely on extra mechanisms to authenticate them. Like the above methods, they do not provide information about the behavior of authenticated nodes.

Some previous work try to solve the security problems based on estimating the behavioral aspects of nodes. For example, *reputation systems* [6, 7]. In general, nodes are rated according to their observed behaviors. The routing decision is then based on the *reputation* of the nodes along the route. However, these works do not address the problem of authentication. Therefore, if a node's reputation becomes bad, it can change its identity and restart its malicious behaviors. Moreover, they do not address the integrity of the reputation information.

Other approaches use a reward and charging scheme to enforce good behavior [9–12]. In these schemes, nodes are assumed to be inherently selfish but rational. This means they will not do anything that will cause significant harm to themselves. However, the assumption that nodes are selfish but rational is only reasonable when all the nodes in the network are resource constrained.

Consider the possibility that some rogue nodes are resource abundant such that the rewards are not attractive to them.

4 Trust Model and Assumptions

In our research, we assume that authenticity is provided by an existing authenticated routing protocol, such as Ariadne. Hence the authenticity of identities, control messages and routes is always assured. We focus on the upper layer of our architecture, risk analysis.

We define the meaning of trust according to the context of routing protocols. We formally define “forwarding trust” in the following.

Definition 1. *Given a route R from S to D and two nodes X and Y on the route, we call X an upstream node of Y (following R) if X is closer to S on the route. Correspondingly, we call Y a downstream node of X (following R).*

Definition 2. *X has forwarding trust of Y (we call X trusts Y in short for the rest of this paper), if and only if the following two conditions hold.*

1. *For all routes R , if X sends or forwards any packets for S following R and Y is a downstream node of X , then with some level of certainty, Y forwards the packets along R unless it is the final destination;*
2. *For all routes R , if X sends or forwards any packets towards D following R and Y is an upstream node of X , Y forwards the packets following R with some level of certainty.*

Informally, when X trusts Y , X believes that Y will forward packets for X , with a certain level of confidence. Note that trust is subjective. X may trust Y although other nodes may distrust Y . Similarly, a malicious node may exhibit malice only to certain set of nodes. Therefore if X trusts Y , the above definition implies that X has a certain level of confidence that Y will also forwards packets even when X is not the source. Since X is committed to forward packets for the source, if Y drops packets, Y is not only exhibiting malicious behavior towards the source, but also towards X . Therefore, by dropping packets originated from the source, Y is also betraying X 's trust. In the event that Y does not wish to forward packets for the source, Y can either choose not to participate in the route discovery or Y can drop the packets and therefore Y betrays X 's trust. The term “some level of certainty” in definition 2 takes into account situations where Y betrays X . This also reflects that there is a risk involve in trusting a node. The same argument can also be applied to condition 2

To address the level of certainty, we need to quantify trust. We map trust from X to Y to a value tr_{XY} between the interval $[0, 1]$, representing the probability that Y will behave well according to definition 2, from X 's point of view. When X trusts Y completely, we have $tr_{XY} = 1.0$. When X distrusts Y definitely (i.e. X thinks Y is a bad node definitely) we have $tr_{XY} = 0$. Between these two extremes, X can assign Y a trust value β in the interval of $(0, 1)$. β depends

on the information (evidence) available. For simplicity, we assume that all the nodes which are neither fully trusted nor fully distrusted have the same β values. This assumption does not harm the generality of the method.

The assignment of values to tr_{XY} is dependent on trust establishment method. Nodes may choose their own system of establishing trust. For example, while some nodes may choose the method in [7], a node who is paranoid may choose to give a neutral rating to nodes with high reputations. It is for this reason that we emphasized that trust is subjective. We do not address trust establishment in this paper. Trust is such a subjective issue that no single solution works for everyone, since everyone has their own notion of trust as illustrated above. Rather than developing our own method to establish trust and arguing its relative merits with existing works, we allow different approaches to be used.

Since not all nodes on a route are trusted, the source node needs to decide which nodes to trust using partial trust relationships. Consider a route (S, A, B, D) , in which S only trusts A . With this partial trust relationship, S needs to determine whether B can be trusted. Suppose A trusts B . Then with some level of certainty, B will forward messages for A . If B drops S 's data packets, B betrays A 's trust. Hence it is implied that S can trust B too. This seems to suggest that forwarding trust is transitive. However, this is not the case. Suppose later on, if S needs to send data to D' using the route (S, B, D') , the relation S trusts B is no longer valid for this route. This is because in the absence of A , B can drop S 's data packets without betraying A 's trust. We call this the property of conditional transitivity.

Definition 3. *The conditional transitivity of trust implies that for all distinct nodes X, Y and Z , if X trusts Y and Y trusts Z , then X trusts Z iff X, Y and Z belongs to the same route.*

Using the property of conditional transitivity, we define a trust chain, with respect to a route, as follows.

Definition 4. *A trust chain, with respect to a route, is a sequence of nodes (X_1, X_2, \dots, X_k) ($k \geq 2$), such that (1) X_i trusts X_{i+1} ($1 \leq i < k$), (2) the sequence preserves either the order or reverse order of the route, and (3) the sequence is not a subset of any other sequences obtained from the same route. As a special case, a sequence of a single node is also a trust chain.*

A trust chain which either starts with S and preserves the order of R or starts with D and preserves the reverse order of R is called a routeable trust chain.

As an example, consider Figure 1 where an arrow going from X to Y represents X trusts Y . The trust chains are: (S, B, C, D) , (D, B, S) and (C, A, S) . However, (S, B) is not a trust chain since it is a subset of (S, B, C, D) . Although (C, A, S) is a trust chain, it is not a routeable trust chain since it does not begin with S or D . To simplify route evaluation, we only consider routeable trust chain in our protocol.

We assume that the source and the destination are always trusted with respect to routing. We do not consider the case in which a destination drops packets

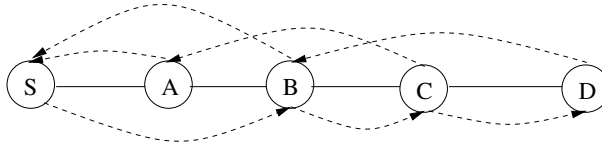


Fig. 1. An example of trust assertions.

irrationally since by doing so, it defeats the purpose of establishing a route for the source and destination to communicate. Note that we are only referring to forwarding trust.

To bootstrap the route evaluation process, we assume the existence of established trusts. For example, trusts exist between nodes who are “friends” with each other, or reputation ratings from existing work can also be used as an initial measurement of trust.

5 Protocol

In this section, we design, as a proof of concept, a routing protocol RBP, using risk as the decision function for route evaluation.

5.1 Overview

RBP is a routing protocol based on DSR [14]. Like DSR, RBP consists of two basic operations: route discovery and route maintenance. Each node also maintains a route cache, which contains routes to various nodes. If there are no routes in the cache to a given destination, a route discovery process is initiated. Route maintenance mainly deals with route errors.

Our ultimate goal in RBP is to improve the throughput in a malicious environment with little additional overhead and delay. A natural way to achieve this is to improve the “quality” of the selected route.

Definition 5. *The quality of a route is the probability that by taking this route, the packet is successfully transmitted to the destination. The quality of a path p is denoted by $q(p)$, and by definition, $q(p) \in [0, 1]$.*

Definition 6. *The candidate path set from the source S to the destination D is the set of all the paths to D in S 's route cache at a given time.*

The quality metric is related to risk. The higher the quality, the lower the risk. The source node aims to reduce the risk of its packets being dropped by choosing routes with high quality. Hence, this is the risk analysis component of RBP. In order to improve throughput, RBP first improves the overall quality of the candidate path set in the route discovery process. Specifically, RBP allows nodes to assert their trust relationships with other nodes on the same route during route discovery. The trust assertions are then used to evaluate the quality

of the route. An intermediate node will only forward a route request if the partial route in the route request is of “good” quality. At the end of route discovery, the source node will the route in the candidate path set that has the highest quality.

5.2 Trust assertion propagation

In the route request and route reply phase, each intermediate node will check the current partial route to see if it has trust relationship with any of the nodes on the partial route. If it does, it will add a *trust assertion* for each of the trusted nodes to the route request packet.

Definition 7. A *trust assertion* $T(X, Y, tr_{XY})$ is a data structure which asserts that X trusts Y with a value of tr_{XY} .

In this paper, unless specified, $tr_{XY} = 1.0$ and $T(X, Y)$ represents $T(X, Y, 1.0)$. $T(X, Y)$ is also represented in diagrams as a dashed arrow going from X towards Y .

In the route request (from S to D), all the trust assertions are from downstream nodes to upstream nodes because a node does not know the downstream nodes at the time when it receives the route request. In the route reply packet, trust assertions added are from upstream nodes to the downstream nodes.

5.3 Route Evaluation

Central to our method is the evaluation of routes. Source node evaluates routes in its candidate path set to choose the route with the best quality. During route discovery, intermediate nodes evaluate partial routes, as described in details later. For this reason, we present route evaluation technique before route discovery. An example route from S to D is given in Figure 1.

We extract all the routeable trust chains. Then all the nodes on these routeable trust chains could be viewed as trusted nodes according to the conditional transitivity of trust property.

For each trusted node, X_i , in the routeable trust chain, we assign a trust value of tr_{X_{i-1}, X_i} or tr_{X_{i+1}, X_i} , depending on whether the routeable trust chain starts from S or D . For unknown nodes, we assign a trust value of β . As an example, in Figure 1, because of valid trust chains (S, B, C, D) and (D, B, S) , B and C can be viewed as trusted while A is unknown. Therefore, S assigns trust values to the nodes as follows: $tr(A) = \beta$, $tr(B) = tr_{SB} = tr_{DB}$, and $tr(C) = tr_{BC}$.

The quality of route $R = (S, X_1, \dots, X_k, D)$ is given by:

$$q(R) = \prod_{i=1}^k tr(X_i) \quad (1)$$

Recall that in our work, we let $tr_{XY} = 1.0$ for any nodes X, Y . Therefore in Figure 1, the quality of the route is β .

Although we assign a trust value of 1.0 to each trusted node in this work, it does not have to be the case. We choose to assign 1.0 to simplify things. In

actual fact, the source node is free to assign any values according to its own policies. Similarly, if S has multiple values to choose from, such as deciding whether $tr(B) = tr_{SB}$ or $tr(B) = tr_{DB}$, the outcome of the decision depends on S 's policies.

5.4 Route Discovery

RBP's route discovery process is similar to that of DSR. However, we perform selective request broadcasting in this process to single out good partial paths.

When S has data to send to the destination D , it first looks up its route cache. If there is no path to D in its route cache, S broadcasts a route request packet:

$$S \rightarrow *: \text{Route_Request}, S, D, seqNo, R' = (S), \\ T_Set = \{\}$$

where $seqNo$ is the sequence number and the current partial route R' , which only contains S at the moment. T_Set represents the set of trust assertions. At this moment T_Set is empty.

When an intermediate node I receives a route request, it first checks whether it has sent out a route request for S with a sequence number greater than $seqNo$. If it has, it discards the route request packet.

If it is the first route request from S with $seqNo$, unlike in DSR, I does not immediately broadcast it because the partial route contained may not have a high enough quality. Also it might be a packet for rushing attacks [3].

Our solution is to buffer the route request for a short time, depending on the quality of the partial route. To compute the quality of the partial route, I first adds trust assertions, if any, to the upstream nodes of R' . It then computes the quality of R' , following Equation (1) as if it were the final destination D .

Let the quality of the partial path be $q(R')$. I buffers the route request for $\alpha(1 - q(R'))$ seconds (called the *request buffer time window*), where α is a system constant⁴.

If during the interval of the request buffer time window, I receives another route request from S with the same sequence number, it checks whether the partial route R'' has higher quality. If it does, the original route request is discarded and the new route request is buffered for $\alpha(1 - q(R''))$ seconds. This process continues until the timer expires. This buffered route request must contain a partial route with the highest quality among all the partial routes from S seen so far. I then broadcasts the expired route request after adding itself to R' .

When a route request arrives at the destination D , it appends itself to the partial route and also adds trust assertions for those nodes on the partial route. Then it reverses the route contained in the route request packet and unicasts a route reply to the source as DSR does. D replies to every route request it receives from S so that S have multiple paths to D . In the process of route reply, nodes

⁴ It is not clear whether a linear formula is the best strategy here. But, our simulations showed a linear request buffer time window works just fine.

will also add trust assertions into T_Set , the set of trust assertions, if they trust some of their downstream nodes.

One attack might be that malicious nodes do not buffer the route request packet it receives and tries to send it out immediately. The consequence is that routes consisting of malicious nodes tends to broadcast faster and S might lose some data packets by following these bad routes when good routes are not available at the moment. These attacks can only succeed in the initial short time window when the bad route arrives at the source while other good routes have not. But after the short time window, since bad routes have lower qualities, they will be ignored. So this attack is greatly limited.

Another potential attack from malicious nodes might be for them to remove trust assertions inserted by other nodes. However, this actually makes the route worse because this potentially reduces the number of trust chains and reduces the number of trusted nodes to the source.

Since we assume the assurance of route authenticity, it is not possible for malicious nodes to fake arbitrary trust assertions for other nodes to increase the quality of the routes. However it can freely assert trust on other nodes. The success likelihood of this attack depends on the trust establishment method, which is out of scope of this paper. A good trust establishment may be such that the probability that a good node trusts a malicious node is very low. Hence the probability that the trust assertions made by the malicious nodes to appear on any valid trust chains is also very low.

5.5 Route Selection

Section 5.4 allows the source S to have multiple paths to D . Rather than selecting the route with the smallest number of hops, we select the route with the highest quality. Only when two or more routes have the same highest quality do we select the route with the smallest number of hops.

5.6 Route Maintenance

All the intermediates nodes as well as the source will have a chance to purge relevant route entries in their route cache according to the route error packet. Note that route errors should also be authenticated so that malicious nodes cannot forge fake route errors. That is, other nodes should have a way to verify that the route error does originate from the node originally finding the link failure.

However, malicious nodes can hide route error packets generated from other good nodes from its downstream so that its upstream nodes do not know the fact. There are three alternatives to address this problem.

First, if the reporting node has multiple routes to the source, it can send the route error packets along multiple paths to the source. A second method is to periodically send a packet to the destination and requires acknowledgment from the destination. However, this method requires a little bit of overhead. The

third method is to attach an age to each route to the destination. We attach a timer with each route. The timeout is inversely proportional to the quality of the route. When all the routes time out, we initiate a new route request process.

6 Evaluation

6.1 Performance

To evaluate the performance of RBP, we implement it in *ns-2* [16]. DSR is not a route authentication protocol but we assume that it is authenticated. We call it DSR_auth in the following. We do not simulate attacking behaviors which exploit authentication related security holes. We present a very simple attacking model here. Malicious nodes will drop data packets and route error messages. They will allow other control messages to pass through.

System Parameters As described previously, in RBP, we need to decide on the following parameters: α and β .

Literally, α represents the maximum buffering time for route request messages. Let the delay between two neighboring nodes be T_D .

In the simulation code, the timeout for ARP is about 0.03s. Thus we can estimate $T_D = 0.015$ (actually this is an upper bound value). As a guideline, α should neither be too much greater than T_D nor too much smaller than T_D . In our simulation, α is 0.04.

The accuracy of β depends on how much extra information (evidence) is available. If we could estimate the fraction of malicious nodes, b , in the system, we could set $\beta = 1 - b$. In general cases, we often assume that the number of malicious nodes are less than good nodes, thus we can use $\beta = 0.5$ in most cases.

Simulation Setting We set 50 nodes in a 670m by 670m area. Each node in our simulation moves according to the *random waypoint* model [15], at a velocity of 20m/s. Nodes transmit at a constant bit rate (CBR) of 4 packets per second, each packet is of 64 bytes.

We are going to compute the following metrics for the performance:

- *Throughput*: The fraction of CBR data packets sent that are received.
- *Delay*: The average time between a CBR data packet is sent and received.

All the above metrics are averaged over 10 runs with different CBR and mobility scenarios. RBP and DSR_auth run on the same scenarios each time.

Simulation Result According to b , the fraction of malicious nodes in the network, we categorize the attacks into three degrees: light attack ($b = 0.1$), medium attack ($b = 0.3$) and severe attack ($b = 0.5$).

In most of the simulations, we will use the following typical parameters to show how RBP performs under attacks compared to DSR_auth: $\alpha = 0.04$ and $\beta = 1 - b$.

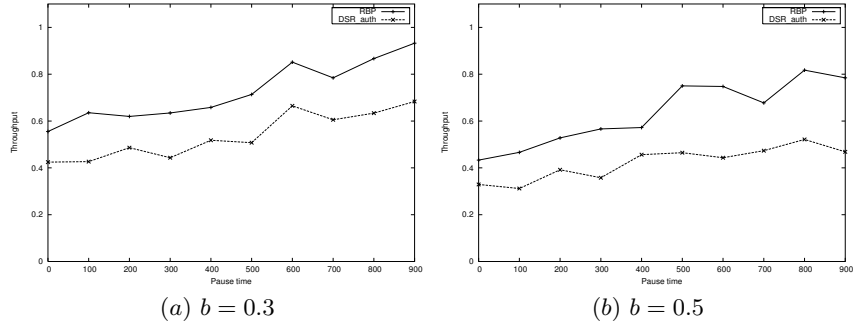


Fig. 2. Throughput comparison in a typical setting for RBP, where $\alpha = 0.04$ and $\beta = 1 - b$.

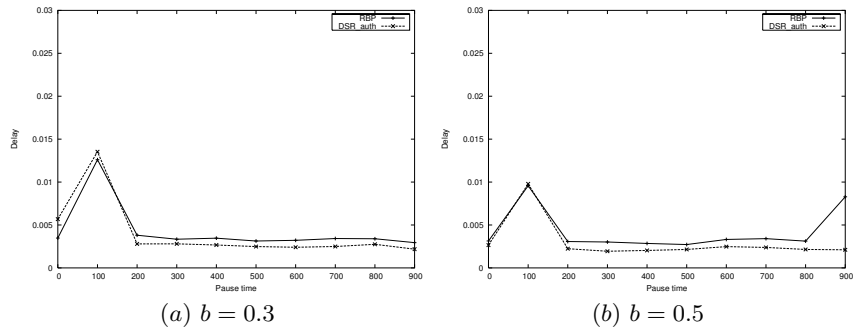


Fig. 3. Delay comparison in a typical setting for RBP, where $\alpha = 0.04$ and $\beta = 1 - b$.

Unfortunately, due to space constraints, we are unable to reproduce every diagrams. In such situations, we will only quote the results we obtained.

Figure 2 and Figure 3 plot the throughput and delay respectively, of the RBP with typical settings, compared to DSR_auth.

We observe that under all three levels of attacks, RBP greatly improves the throughput while involving only trivial delays.

7 Conclusions

We have proposed a secure architecture for routing in ad hoc networks. Our architecture emphasizes the importance of authenticity and behavior of nodes. It is our hope that this architecture will promote the awareness of the coupling relationship between these two issues, since previous work only addresses these issues in isolation.

We have also proposed RBP. One of the highlights of RBP is through the use of partial information in the form of trust chains, RBP computes the probability

of successful delivery of data along a given route. In the absence of complete and reliable information, the best that a node can do is to make use of the partial information it has to improve the delivery rate.

Our simulation shows that RBP increases throughput in a malicious environment. Results show that throughput is increased by at least ten percent over non-optimized DSR even when fifty percent of the nodes in the network are malicious. The increase in throughput is achieved with only a small delay.

References

1. Y.-C. Hu, A. Perrig and D. B. Johnson. Ariadne: a secure on-demand routing protocol for ad hoc networks. *MobiCom 2002*.
2. Y.-C. Hu, D. B. Johnson and A. Perrig. Secure efficient distance vector routing in mobile wireless ad hoc networks. In *Proceedings of the 4th IEEE Workshop on Mobile Computing Systems and Applications (WMCSA)*. June 2002.
3. Y.-C. Hu, A. Perrig and D. B. Johnson. Rushing attacks and defense in wireless ad hoc network routing protocols. *WiSe2003*.
4. M. G. Zapata and N. Asokan. Securing ad hoc routing protocols. *WiSe 2002*.
5. K. Sanzgiri and B. Dahill. A secure routing protocol for ad hoc networks. *ICNP 2002*.
6. S. Buchegger and Jean-Yves Le Boudec. Performance Analysis of the CONFIDANT Protocol (Cooperation Of Nodes: Fairness In Dynamic Ad-hoc Networks). *MobiHoc2002*.
7. Sergio Marti and T. J. Giuli and Kevin Lai and Mary Baker. Mitigating routing misbehavior in mobile ad hoc networks. *MobiCom 2000*.
8. John R. Douceur. The sybil attack. In *Proceedings of the 1st International Workshop on Peer-to-Peer Systems (IPTPS '02)*, 2002.
9. L. Buttyan and J.-P. Hubaux. Enforcing service availability in mobile ad hoc WAnS. In *Proceedings of the IEEE/ACM Workshop on Mobile Ad Hoc Networking and Computing (MobiHOC)*, Boston, August 2000
10. L. Buttyan and J.-P. Hubaux. Stimulating cooperation in self-organization mobile ad hoc networks. *ACM/Kluwer Mobile Networks and Applications*, 8(5). Oct 2003.
11. M. Jakobsson, J.-P. Hubaux and L. Buttyan. A micro-payment scheme encouraging collaboration in multi-hop cellular networks. In *Proceedings of Financial Cryptography*, 2003
12. S. Zhong, Y. R. Yang and J. Chen. Sprite: a simple, cheat-proof, credit-based system for mobile ad hoc networks. In *Proceedings of INFOCOM*. IEEE, 2003
13. Srjjan Capkun and J.-P. Hubaux. BISS: building secure routing out of an incomplete set of security associations. *WiSe 2003*.
14. D. Johnson, D. Maltz and J. Broch. DSR The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks. In *Ad Hoc Networking, edited by Charles E. Perkins*, Chapter 5, pp. 139–172. Addison-Wesley, 2001.
15. J. Broch, D. A. Maltz, D. B. Johnson, Y.-C. Hu and J. Jetcheva. A performance comparison of multi-hop wireless ad hoc network routing protocols. In *Proceedings of the 4th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MOBICOM'98)*, Dallas, TX, USA, October 1998, pages 85-97.
16. The Network Simulator ns-2. <http://www.isi.edu/nsnam/ns/>