Noname manuscript No.
(will be inserted by the editor)

Charikleia Zouridaki · Brian L. Mark* · Marek Hejmo

Byzantine Robust Trust Establishment for Mobile Ad hoc Networks

Received: date / Revised: date

Abstract In a mobile ad hoc network (MANET), the nodes act both as traffic sources and as relays that forward packets from other nodes along multi-hop routes to the destination. Such networks are suited to situations in which in a wireless infrastructure is unavailable, infeasible, or prohibitively expensive. However, the lack of a secure, trusted infrastructure in such networks, make secure and reliable packet delivery very challenging. A given node acting as a relay may exhibit Byzantine behavior with respect to packet forwarding, i.e., arbitrary, deviant behavior, which disrupts packet transmission in the network. For example, a Byzantine node may arbitrarily choose to drop or misroute a certain percentage of the packets that are passed to it for forwarding to the next hop. In earlier work, we proposed a trust establishment framework, called Hermes, which enables a given node to determine the "trustworthiness" of other nodes with respect to reliable packet delivery by combining first-hand trust information obtained independently of other nodes and second-hand trust information obtained via recommendations from other nodes. A deficiency of the Hermes scheme is that a node can fail to detect certain types of Byzantine behavior, such as packet misforwarding directed at a particular source node. In this paper, we propose new mechanisms to make Hermes robust to Byzantine behavior and introduce a punishment policy that discourages selfish node behavior. We present simulation results that demonstrate the effectiveness of the proposed scheme in a variety of scenarios involving Byzantine nodes that are malicious both with respect to packet forwarding and trust propagation.

Keywords Mobile ad hoc network, Trust Establishment, Byzantine Behavior, Network Security, Secure Routing

1 Introduction

In a mobile ad hoc network (MANET), the nodes in the network act both as traffic sources and as relays that forward packets from other nodes along multi-hop routes to the destination. Such networks are suited to situations in which in a wireless infrastructure is unavailable, infeasible, or prohibitively expensive. However,

C. Zouridaki · B.L. Mark · M. Hejmo
Dept. of Electrical and Computer Engineering, George Mason University, MS 1G5, Fairfax, VA 22030, U.S.A.
E-mail: {czourida, bmark, mhejmo}@gmu.edu

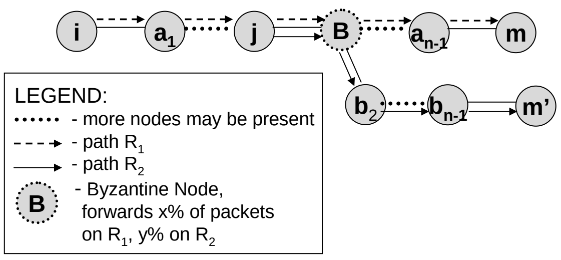

i
a1
j
B
an-1
m
b2
bn-1
m'
LEGEND:
- more nodes may be present
- path R1
- path R2
- Byzantine Node, forwards x% of packets on R1, y% on R2

Fig. 1 Example scenario illustrating the Byzantine node problem.

the lack of a secure, trusted infrastructure in such networks, make secure and reliable packet delivery very challenging. A given node acting as a relay may exhibit Byzantine behavior [12,1] with respect to packet forwarding, i.e., arbitrary, deviant behavior, which disrupts the normal packet forwarding function of the network. For example, a Byzantine node may arbitrarily choose to drop or misroute a certain percentage of the packets passed to it that are to be forwarded to the next hop along a route.

Motivated by the need to secure MANETs, there has been considerable recent interest in the topic of trust establishment for ad hoc networks [6,8,3,13,14,4,5,9,2]. In earlier work, we proposed a trust establishment framework, called *Hermes* [16], which enables a given node to determine the "trustworthiness" of other nodes with respect to reliable packet delivery by combining first-hand trust information obtained independently of other nodes and second-hand trust information obtained via recommendations from other nodes. First-hand trust information with respect to neighbor nodes is obtained by gathering statistics on packet forwarding behavior observed at the MAC (medium access control) layer. Second-hand trust information with respect to non-neighbor nodes is obtained through the exchange of recommendations. The *E-Hermes* (Extended-Hermes) scheme, proposed in [17], extends the original Hermes scheme by enabling nodes to gather first-hand trust information with respect to non-neighbor node via a secure acknowledgement scheme. The acknowledgement scheme results in a more robust trust establishment scheme, which can thwart nodes that attack the trust establishment scheme itself by propagating erroneous recommendations.

A deficiency of the E-Hermes scheme [17] is that a node can still fail to detect certain types of Byzantine behavior, i.e., behavior that can deviate in an arbitrary manner [12,1]. In particular, the E-Hermes scheme is vulnerable to attacks by nodes that misforward data packets in a selective manner. For example, a Byzantine node may choose to drop packets belonging to a certain set of source nodes, or to forward only packets belonging to another set of source nodes. Consider the example scenario shown in Fig. 1. Let

$$R_1 = \{i, a_1, ..., a_j = j, a_{j+1} = B, ..., a_{n-1}, a_n = m\},$$

denote a path from node i to node m, and

$$R_2 = \{j, b_1 = B, b_2, ..., b_{n-1}, b_n = m'\},$$

denote a path from node j to node m'. Note that node j is an intermediate node on the path R_1 and the source node of the traffic flow that traverses path R_2. Suppose that node B exhibits Byzantine behavior by *correctly* forwarding all the packets that node i sends to it on path R_1, but *incorrectly* forwarding 20% of the packets that node j sends to it on path R_2. Assume that all other other nodes forward all packets *correctly.*

If both source nodes i and j send 100 packets for forwarding to their respective destinations, node j observes that node B *correctly* forwards 120 of the 200 packets that it received for forwarding. Node j's goal is to determine which nodes it should "trust" for forwarding its data packets to their destination nodes. If node j does not differentiate between the empirical evidence that it accumulates as source from the empirical evidence that it accumulates as intermediate node, it will not be able to identify which nodes exhibit Byzantine behavior for its flows. Under the E-Hermes framework, node j only observes that node B forwarded 120 out of 200 packets. Hence, node j is unable to recognize that node B is exhibiting Byzantine behavior. This suggests that node j should rely mainly on observations of node forwarding behavior with respect to its own packets, i.e., 20 out of the 100 packets it forwards in this example.

While the E-Hermes scheme provides a means for a node to determine the trustworthiness of other nodes, it does nothing to discourage nodes from acting selfishly. For example, a node could simply drop all packets forwarded to it in order to conserve its own battery power. Under E-Hermes, such a node would be identified as a "bad" node by other nodes, but it would have no incentive to alter its behavior if its primary objective were to conserve battery power. Thus, a trust establishment scheme alone is not sufficient to alleviate the effects of Byzantine node behavior.

In this paper, we propose new mechanisms that make the Hermes scheme robust to Byzantine behavior. We also introduce a punishment policy that discourages selfish node behavior. We refer to the new trust establishment scheme, including the punishment policy, as *Byzantine Robust Hermes* or *BR-Hermes*, to distinguish it from the earlier Hermes and E-Hermes schemes mentioned above. The main difference between BR-Hermes and E-Hermes is that in the BR-Hermes scheme, each node distinguishes whether it is a source node or intermediate node with respect to a given packet flow that provides first-hand information on node behavior. This information is then used to compute a first-hand trust metric that is robust to Byzantine node behavior. An additional benefit of the BR-Hermes scheme is that its property to identify Byzantine behavior can be exploited to develop a a punishment policy that discourages nodes from selfishly dropping packets. The main features provided by the BR-Hermes scheme are summarized as follows:

– ability to determine accurate trustworthiness information for any node in the network in the presence of Byzantine nodes;
– responsiveness to changes in node behavior;
– ability to distinguish between malicious behavior with respect to packet forwarding vs. trust propagation;
– ability to identify the effect of attacks by individual or colluding malicious nodes.

The remainder of the paper is organized as follows. Section 2 briefly discusses related work. Section 3 provides an overview of the concepts underlying the Hermes family of trust establishment schemes. Sections 4 and 5 discuss the core concepts and advances of the paper. Section 4 presents a Byzantine robust scheme for accumulating trust information and computing trust metrics for neighbor and non-neighbor nodes. Section 5 introduces a punishment scheme that discourages selfish node behavior. In Section 6, we present a security evaluation of our BR-Hermes trust establishment scheme. Section 7 presents results from simulation experiments that demonstrate the performance properties of BR-Hermes. In particular, we provide numerical comparisons of the BR-Hermes scheme versus the E-Hermes scheme, which does not detect Byzantine behavior. The overhead of incurred by BR-Hermes is also discussed in Section 7. Finally, the paper is concluded in Section 8.

2 Related Work

The authors of [6] present a high-level framework for generation, revocation and distribution of trust evidence and demonstrate the significance of estimation metrics in trust establishment. A mechanism for trust evidence dissemination based on a model of ant behavior is proposed in [8] along the lines suggested in [6]. Others have approached trust establishment based on the use of a Bayesian framework [3,15]. The Bayesian approach was initially explored in [3]. The Hermes scheme presented in [16] builds on the Bayesian approach by incorporating the notion of statistical confidence associated with a trust value.

In [13], a trust model is presented that allows the evaluation of the reliability of the routes, using only first-hand information. The notion of confidence as it related to trust management was explored in [14] and a semi-ring approach was suggested to evaluate trust and confidence along network paths. The Hermes framework for trust management introduced in [15,16] maps trust and confidence into a new composite metric, called “trustworthiness,” which can be more easily used for making network decisions such as route selections. Furthermore, Hermes deals directly with the issue of how evidence can be collected from the network to establish and update trust.

The E-Hermes scheme [17] extends the original Hermes scheme by addressing an attacker model where nodes can exhibit malicious behaviors independently, i.e. failure to forward packets is independent of the honesty with which trustworthiness values are propagated about other nodes. Another extension over Hermes is that more accurate trustworthiness values for non-neighbor nodes based on first-hand information fvia an acknowledgement scheme, as opposed to relying only on second-hand recommendations. In E-Hermes, recommendations are still used to promote faster convergence of the trust establishment procedure and to model the trustworthiness of third-party recommender nodes.

3 Hermes Trust Establishment Framework

In this section, we set the stage for the rest of the paper by giving a brief overview of the Hermes trust management framework introduced in [15,16] and the extended version of Hermes called E-Hermes, proposed in [17].

3.1 Overview of Trust Management Concepts

The notion of trust and trust relationships have been studied extensively in the literature [10]. Associated with the notion of trust is confidence, which is a measure of the level of assurance in the trust relationship. It is helpful to combine trust and confidence into a composite notion called *trustworthiness* [15] as it makes trust-related computations more straightforward. We apply all these notions to the problem of reliable packet delivery in MANETs. First-hand information on packet delivery is what can be directly observed by the sender in a path, but second-hand information can only be obtained from third-parties. The literature discusses the conveyance of second-hand information through a variety of schemes such as recommendations. In Hermes [15,16], opinions represent the combination of first-hand and second-hand information, the latter being gathered through recommendations.

We briefly review the notions of trust, confidence, and trustworthiness introduced in the Hermes scheme. Consider a given node that is observed over time with respect to its packet forwarding behavior. Let A denote the cumulative number of packets forwarded correctly and let M denote the cumulative number of packets forwarded incorrectly by the node up to the current time. Then the trust value, t, assigned to a node is defined as follows:

$$t \triangleq \frac{A}{M}, \tag{1}$$

where $0 \leq t \leq 1$. A value of t equal to one indicates absolute trust, whereas a value close to zero indicates low trust. This definition of trust is based on Bayesian statistics [15]. The confidence value, c, associated with the trust value t is defined as follows:

$$c \triangleq 1 - \sqrt{\frac{12A(M-A)}{M^2(M+1)}}, \tag{2}$$

where $0 \leqslant c \leqslant 1$. A value of c close to one indicates high confidence in the accuracy of the computed trust value t, whereas a value close to zero indicates low confidence. At instant k a given node can be characterized by a pair (t, c). In particular, node i characterizes its trust in node j by the pair $(t_{i,j}, c_{i,j})$.

The notion of *trustworthiness* was introduced in Hermes to characterize a pair (t, c) of trust and confidence values into a single metric to facilitate trust-based decisions. The trustworthiness associated with a pair (t, c) is defined as [15]:

$$T(t,c) \triangleq 1 - \frac{\sqrt{(t-1)^2 + r^2(c-1)^2}}{\sqrt{1+r^2}}, \tag{3}$$

where r is a parameter that determines the relative importance of the trust value t vs. the confidence value c.

The "default" value of trustworthiness is defined as

$$T_{def} \triangleq T(0.5, 0),$$

which represents the trustworthiness value assigned to a node when its assigned trust and confidence values are $t = 0.5$ and $c = 0$, respectively. Thus, the value T_{def} represents ignorance about the trustworthiness of a node. The value T_{def} can be interpreted as an initial threshold for trustworthiness. If the trustworthiness of a node exceeds T_{def}, then the node is considered "trustworthy" or"good." Otherwise, the node is viewed as "untrustworthy" or "bad." In addition to T_{def}, we also define c_{acc} as an acceptability threshold. To meet c_{acc} the constituent confidence has to be at least ϵ , where ϵ is predefined. Each node may choose a different value of c_{acc} to implement its own policy in determining the acceptability of trustworthiness values.

3.2 First-hand Trust Evaluation

In the Hermes scheme, first-hand trust information for *neighbor* nodes is obtained via direct observations of packet forwarding behavior at the MAC layer. The E-Hermes scheme enables nodes to gather first-hand information with respect to *non-neighbor* nodes via an acknowledgement scheme. The receipt by an acknowledgement (ACK) sent by the destination of a given data flow indicates that a data packet has been received by the destination and thus all the nodes on the path to the destination forwarded the packet to their downstream node. All nodes on the path initiate a timer when they forward a packet and expect to receive a corresponding acknowledgement with the timeout period, which should be larger than the maximum round-trip propagation time along the given path in the network. The handling of dropped ACKs and the creation of negative acknowledgement (NACK) packets is discussed in detail in [17].

Counters are maintained to calculate the number of packets that downstream nodes forward. Observation data is accumulated only for downstream nodes, since an intermediate node knows the number of packets it receives for forwarding from its upstream node, but is unaware of the number of packets that its upstream node had received for forwarding. We also remark that both of the end nodes of a given link identified as "faulty" via receipt of a NACK are penalized. However, this effect becomes negligible as a more diverse set of observation data involving the two nodes is accumulated in the network over time.

3.3 Formulation of Opinions

The concept of *opinion* was introduced in the Hermes framework [15], to generalize the notion of trustworthiness to non-neighbor nodes. In the Hermes scheme, the opinion that a node has for a neighbor node is equivalent to the first-hand trustworthiness metric. The opinion for a non-neighbor node, is obtained via a recommendation from a neighbor node, i.e., a second-hand trustworthiness metric. A drawback of the Hermes definition of opinion metric for non-neighbor nodes is that it renders the scheme vulnerable to misbehaving nodes that propagate erroneous recommendations.

The presence of *bad recommenders* is addressed in the E-Hermes scheme [17], via the acknowledgement scheme mentioned above. Under the acknowledgement scheme, "first-hand" trustworthiness information can be obtained with respect to *non-neighbor* nodes. Thus, the opinion metric is made robust with respect to bad recommenders. Nevertheless, the concept of recommendations is retained in the E-Hermes scheme to accelerate the convergence of the opinion metric. By comparing the first-hand trustworthiness for a given non-neighbor node x with a recommendation for node x received from node y, an opinion metric can be computed with respect to the trustworthiness of node y as a *recommender*. Thus, the E-Hermes scheme is able to distinguish not only packet forwarding misbehavior, but also trust propagation misbehavior.

In the E-Hermes scheme, the trustworthiness of the recommenders j belonging to a set of recommenders D is evaluated by performing the following *recommender's test* or *RC-test* [17]:

$$\text{RC-test} : |T_{i,m} - T_{j,m}| \leq \eta,$$

where $\eta \in (0, 1)$ is a threshold value. The RC-test succeeds when the recommended trustworthiness value is close to the first-hand trustworthiness value as defined by the set threshold. Otherwise, the test fails. The outcome of each RC-test for recommender j is used to update counters A and M, where A counts the number of times for which the RC-test succeeds and M counts the total number of times the RC-test is executed in the current observation window. The A and M counters are then used to calculate the *recommender trustworthiness* $T^R_{i,j}$ according to the trustworthiness formulas (1)- (3). Recommender trustworthiness $T^R_{i,j}$ is the trustworthiness that node i places on recommender node j with respect to reliable propagation of trustworthiness T. Note that a node j declines to submit a recommendation for node m to node i when m is the FIN node of j and $\eta \cdot 100\%$ of the control packets sent from m to j for a given flow are NACKs [17].

We now provide the definition for the opinion, $P_{i,m}$, that any node i has for another node m as follows [17]:

$$P_{i,m} \triangleq \max_{j \in \Gamma}\{\omega_{i,j} T_{j,m}\}, \text{ for } P_{j,m} \neq T_{def}, \tag{4}$$

where

$$\omega_{i,j} = \begin{cases} T^R_{i,j}, \ i \neq j, \\ 1, \quad i = j. \end{cases} \tag{5}$$

and Γ is the set of recommenders in D that pass the RC-test. The opinion $P_{i,m}$ is recalculated as the maximum of the current opinion $P_{i,m}$ and the recommendations in the set Γ, weighted by the recommender trustworthiness value $T^R_{i,j}$.

Authentication of every data, recommendation, ACK and NACK packet is required to protect the network against modification and impersonation attacks. The appropriate authentication mechanisms are discussed in [17]. We remark that when a secure routing algorithm is in place (cf. [11,7,1]), the nodes have already established pairwise keys, which can also be used for the authentication of the information exchanged during the trust establishment phase of Hermes.

4 Trust Evaluation for Byzantine Detection

In this section, we present the BR-Hermes scheme for gathering first-hand trust information from neighbor and non-neighbor nodes. The first-hand information that a node obtains from a traffic flow is weighted depending on whether the given node is the source or an intermediate node on the route that the flow traverses. This is an extension over Hermes and E-Hermes, which gather first-hand information without considering the role of the node on the route. The BR-Hermes scheme requires authentication mechanisms for both data and control packets. The authentication mechanisms discussed in [17] can be applied here.

4.1 Problem Statement

Consider a very simple route $\{x, y, z\}$. In E-Hermes scheme, a given node x in the network maintains counters M_y and A_y for node y. We refer to the sets of counters $\{M_y\}$ and $\{A_y\}$ as M-counters and A-counters, respectively. The counter M_y records the total number of packets sent from node x to node y for forwarding to z over an observation window. The counter A_y records the total number of packets forwarded *correctly* (not dropped or misrouted) from node y to node z. Then the trust and confidence that x attributes to y over an observation window[1] are given by (cf. (1) and (2))

$$t_{x,y} = t(A_y, M_y) \text{ and } c_{x,y} = c(A_y, M_y),$$

from which the trustworthiness value $T_{x,y}$ can be computed via (3).

With respect to the scenario of Fig. 1, we discussed the significance of the empirical evidence that a node collects as a source node in a network. However, this should not minimize the importance of the empirical evidence that can be accumulated when a node serves as an intermediate node on paths in the network. To better understand this point, reconsider the scenario of Fig. 1, with the difference that now node B forwards

[1] Windowing methods to expire old observation data are discussed in [15] as part of the Hermes framework.

correctly all the packets that node i sends to it on path R_1, and all the packets that node j sends to it on path R_2. If both source nodes i and j send 100 packets for forwarding, node j observes that node B forwards *correctly* all 200 packets that it received for forwarding.

Taking these remarks into consideration, we proceed to develop the BR-Hermes scheme, which distinguishes the empirical evidence collected as a source and the empirical evidence collected as an intermediate node and combines them to compute a Byzantine robust trust metric.

4.2 Byzantine Robust Processing of Empirical Evidence

In this section, we present a new scheme for processing the first-hand trust information from neighbor and non-neighbor nodes. The key point of our new scheme is that a node differentiates between the empirical evidence that it collects when it is the source of traffic flows and the empirical evidence that it collects when it is an intermediate node on routes that flows of other sources traverse.

4.2.1 Weighting Empirical Evidence

Let $R = \{a_0, a_1, a_2, \cdots, a_{n-1}, a_n\}$, where $n \geq 2$, denote a path from node a_0 to node a_n. In the BR-Hermes scheme, the source node a_0 maintains counters

$$\widetilde{M}_{a_i} \quad \text{and} \quad \widetilde{A}_{a_i}, \quad 1 \leq i \leq n-1.$$

for its downstream nodes. The intermediate nodes a_j, $1 \leq j \leq n-2$ maintain counters

$$\widehat{M}_{a_i} \quad \text{and} \quad \widehat{A}_{a_i}, \quad j+1 \leq i \leq n-1.$$

for their downstream nodes.

We refer to the sets of counters $\{\widetilde{M}_{a_i}\}$ and $\{\widetilde{A}_{a_i}\}$, $1 \leq i \leq n-1$ as $\widetilde{M}$-counters and $\widetilde{A}$-counters respectively. Similarly, we refer to the sets of counters $\{\widehat{M}_{a_i}\}$ and $\{\widehat{A}_{a_i}\}$, $1 \leq j \leq n-2$, $j+1 \leq i \leq n-1$, as $\widehat{M}$-counters and $\widehat{A}$-counters respectively. The counters M record the total number of packets sent for forwarding over an observation window. The counters A record the total number of packets forwarded *correctly* (i.e., not dropped or misrouted) from the downstream nodes. The number of packets forwarded *incorrectly* by the downstream nodes is given by $B \triangleq M - A$.

- $\widetilde{M}$-counters and $\widetilde{A}$-counters indicate the empirical evidence collected from the network when the node is the source of the traffic flow;
- $\widehat{M}$-counters and $\widehat{A}$-counters indicate the empirical evidence collected from the network when the node is an intermediate node on the path that the traffic flow traverses.

Then the trust $\widetilde{t}_{a_0,a_i}$ and confidence $\widetilde{c}_{a_0,a_i}$ that the source node a_0 attributes to its downstream nodes a_i, $1 \leq i \leq n-1$ over an observation window are given by (cf. (1) and (2))

$$\widetilde{t}_{a_0,a_i} = t(\widetilde{A}_{a_i}, \widetilde{M}_{a_i}), \quad 1 \leq i \leq n-1$$

and

$$\widetilde{c}_{a_0,a_i} = c(\widetilde{A}_{a_i}, \widetilde{M}_{a_i}), \quad 1 \leq i \leq n-1$$

from which the trustworthiness value $\widetilde{T}_{a_0,a_i}$, $1 \leq i \leq n-1$ can be computed via (3).

Similarly, the trust $\widehat{t}_{a_j,a_i}$ and confidence $\widehat{c}_{a_j,a_i}$ that intermediate node a_j, $1 \leq j \leq n-2$, attributes to its downstream nodes a_i, $j+1 \leq i \leq n-1$ over an observation window are given by (cf. (1) and (2))

$$\widehat{t}_{a_j,a_i} = t(\widehat{A}_{a_i}, \widehat{M}_{a_i}), \quad 1 \leq j \leq n-2, \quad j+1 \leq i \leq n-1$$

and

$$\widehat{c}_{a_j,a_i} = c(\widehat{A}_{a_i}, \widehat{M}_{a_i}), \quad 1 \leq j \leq n-2, \quad j+1 \leq i \leq n-1$$

from which the trustworthiness value $\widehat{T}_{a_j,a_i}$, $1 \leq j \leq n-2$, $j+1 \leq i \leq n-1$ can be computed via (3).

The next step is to define a method, which will allow every node to take advantage of the empirical evidence that it collects as intermediate node, without risking to lose its visibility of the network nodes' forwarding behavior for its own traffic flows. In other words, we wish to assign a weight w to the empirical evidence that a nodes collects as an intermediate node, where $0 \leq w \leq 1$. A value of w close to one indicates that the given evidence weights by 100%, whereas a value close to zero indicates that the given evidence does not weight at all, and thus is ignored. When the node is the source of a flow, the information it collects from the network is weighted by $w = 1$.

To decide on the weight w that will be assigned to the information collected by a node when it is an intermediate node on the route that the flow traverses, we calculate the absolute value of the difference between the trustworthiness value $\widetilde{T}_{i,m}$ that node i forms for a node m and the trustworthiness value $\widehat{T}_{i,m}$ that node i forms for a node m:

$$\text{T-Distance} : |\widetilde{T}_{i,m} - \widehat{T}_{i,m}| = e \tag{6}$$

where $e \in [0,1]$, since $\widetilde{T}_{i,m}, \widehat{T}_{i,m} \in [0,1]$. Then, the weight w corresponding to the counters $\widehat{A}$ and $\widehat{B}$ is assigned as

$$w = 1 - e. \tag{7}$$

The smaller the value of the T-Distance, the more node i comes to believe that node m does not exhibit Byzantine behavior, since it behaves similarly when it forwards data packets for different sources. In this case, node i weights the information that it collects as intermediate node more, as defined by (7) and (6). The larger the T-Distance value, the more node i comes to believe that node m exhibits Byzantine behavior, since it behaves differently when it forwards data packets for different sources. In this case, node i weights the information that it collects as intermediate node less, as defined by (7) and (6). If node i has computed a value for $\widehat{T}_{i,m}$, but has not yet computed a value for $\widetilde{T}_{i,m}$, we assume that $e = 0$ and, consequently, $w = 1$.

4.2.2 First-hand Trust Evaluation Scheme

A node i calculates first-hand trust with respect to another node m using counters A_m and M_m. The counters A_m represent the estimated overall number of packets forwarded *correctly* (not dropped or misrouted) from node m over an observation window and thus are given as functions of the counters $\widetilde{A}_m$ and $\widehat{A}_m$. Here, the $\widetilde{A}$-counters indicate the number of packets forwarded *correctly* when the node is the source of the traffic flow, whereas the $\widehat{A}$-counters indicate the number of packets forwarded *correctly* when the node is an intermediate node on the path that a traffic flow traverses.

Similarly, counters M_m represent the total overall number of packets sent for forwarding by node m over an observation window and thus are given as a functions of the counters $\widetilde{M}_m$ and $\widehat{M}_m$, where the $\widetilde{M}$-counters indicate the total number of packets sent for forwarding when the node is the source of the traffic flow and the $\widehat{M}$-counters indicate the total number of packets sent for forwarding when the node is an intermediate node on the path that a traffic flow traverses. Then, the counters A_m and M_m are defined, respectively, as follows:

$$A_m \triangleq \sum_u \widetilde{A}_{m,u} + \sum_v w_v \widehat{A}_{m,v} - 1, \tag{8}$$

and

$$M_m \triangleq \sum_u \widetilde{M}_{m,u} + \sum_v w_v \widehat{M}_{m,v} - 1, \tag{9}$$

where u ranges over the set of flows for which node i is the source node and v ranges over the set of flows for which node i is an intermediate node[2]. Here, the weights w_v are chosen in the range $[0, 1]$, as discussed in Section 4.2.1 Finally, the trust $t_{i,m}$ and confidence $c_{i,m}$ that node i attributes to node m over an observation window are computing using (1) and (2), respectively, as follows:

$$t_{i,m} = t(A_m, M_m)$$

and

$$c_{i,m} = c(A_m, M_m),$$

from which the trustworthiness value $T_{i,m}$ is computed via (3).

4.3 Recommendations

The recommendation scheme introduced as part of the E-Hermes scheme in [17] is adopted in the BR-Hermes scheme. The proposed scheme exploits information sharing among nodes to accelerate the convergence of trust establishment procedures, yet is robust against the propagation of false trust information by malicious nodes. In Section 7, we compare the convergence of BR-Hermes with and without the use of recommendations under different scenarios.

5 Punishment Scheme

The ability of the BR-Hermes scheme to identify Byzantine behavior enables us to develop a punishment scheme for nodes that do not forward correctly the packets that they receive for forwarding. Such a scheme is needed to prevent selfish nodes from dropping packets to ensure that they are not used as intermediate nodes, hence saving their battery power.

Assume that network node i computes at a given time instance $t = k$ opinion $P_{i,m}$ for node m (see (4)). Let $B_f^{i,m}$, where $0 \leq B_f^{i,m} \leq 1$, denote the probability that the node i performs incorrect packet forwarding

[2] The subtraction of 1 on the right-hand sides of (8) and (9) ensures that the counters A_m and M_m have the correct values at system initialization time.

for node m in steady-state. We can extend the relationship between two nodes to define a punishment scheme for nodes. More precisely, we define the behavioral policy for node i as follows:

$$B_f^{i,m} = \begin{cases} 0, & P_{i,m} = T_{def}, \\ 1 - P_{i,m}, & \text{otherwise.} \end{cases} \tag{10}$$

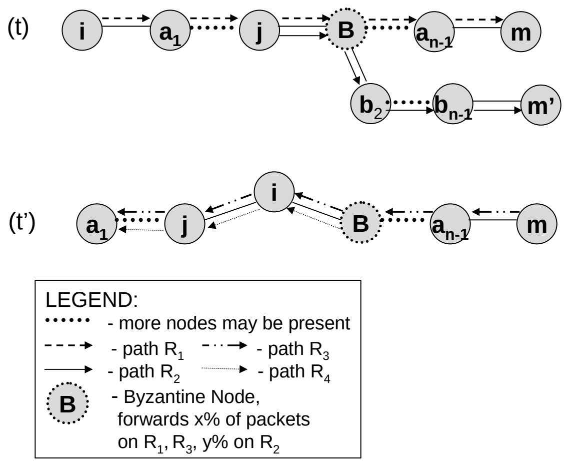
Under BR-Hermes, node i chooses its forwarding behavior based on its observations and computation of its opinion $P_{i,m}$ for another network node m. At initialization time, node i does not have an opinion for node m, and until it forms its opinion, node i is required to forward correctly the packets it receives from node m for forwarding. When node i forms its opinion $P_{i,m}$ for node m, node i will base its forwarding behavior on this opinion as given by (10). We note that a Byzantine node may not respect this policy and may base their forwarding behavior on an arbitrary set of rules.

The behavioral policy defined by (10) can be viewed as a punishment scheme, since a node's packet forwarding behavior would be expected to be characterized by $B_f^{i,m} = 0$ at any time. However, it is important to punish nodes that attempt to thwart the trust establishment scheme by dropping or misrouting packets in order to ensure that they will not used as intermediate nodes in paths. Therefore, in BR-Hermes, node i is expected to forward packets *incorrectly* (drop or misroute) according to (10) to mitigate such malicious behavior. We note that for a trust establishment scheme to be effective, it must also be capable of adapting to the dynamic changes in the node behavior and the network topology. This issue is addressed in the Hermes scheme [15,16] via the use of windowing mechanisms.

We do not define our behavioral policy by $B_f^{i,m} > 1 - P_{i,m}$ (when $P_{i,m} \neq T_{def}$), in which case node i would forward *incorrectly* for node m more packets than node m forwards *incorrectly* for node i. This would result in a vicious circle, where one node would forward more and more packets *incorrectly* for the other node in order to satisfy the behavioral policy and eventually, both nodes would not forward correctly any packets for each other, causing a denial of service condition. Moreover, we do not define our behavioral policy as $B_f^{i,m} < 1 - P_{i,m}$ (when $P_{i,m} \neq T_{def}$), where node i would forward *incorrectly* for node m less packets than node m forwards *incorrectly* for node i. Such a measure would not discourage node m from forwarding *incorrectly* packets for node i, since node i would continue forwarding more packets for node m than node m would forward for node i.

Consider the scenario shown in Fig. 2, which is similar to the scenario of Fig. 1. Here, node j is an intermediate node on the path R_1 and the source node of the traffic flow that traverses path R_2. Node B is an intermediate node on both paths and exhibits Byzantine behavior. In particular, node B forwards *correctly* all the packets that node i sends to it on path R_1, whereas it forwards *correctly* only 20% of the packets that node j sends to it on path R_2. The other nodes forward all the packets *correctly.*

Under the BR-Hermes scheme, the intermediate nodes on the paths follow the behavioral policy (10). Since at initialization time, all opinions equal T_{def}, all nodes should perform incorrect packet forwarding with probability zero. In the scenario of Fig. 2, assume that all nodes follow this policy, except for node B, which exhibits Byzantine behavior. At time t, node i forms opinion $P_{i,j}(t)$ for node j based on node's j *overall* forwarding behavior and node j forms opinion $P_{j,B}(t)$ for node B based on node's B *overall* forwarding behavior (as intermediate node on node's i flow on path R_1 and node's j flow on path R_2).



(t)
i
a1
j
B
an-1
m
b2
bn-1
m'
(t')
i
a1
j
B
an-1
m
LEGEND:
- more nodes may be present
- path R1
- path R3
- path R2
- path R4
B
- Byzantine Node, forwards x% of packets on R1, R3, y% on R2

Fig. 2 Illustration of punishment scheme in BR-Hermes.

Suppose[3] that at time t',

$$R_3 = \{a_n = m, a_{n-1}, ..., a_{j+1} = B, a_0 = i, a_j = j, ..., a_1\},$$

where $n \geq 4$, denote a path from node m to node a_1, and

$$R_4 = \{a_{j+1} = B, a_0 = i, a_j = j, ..., a_1\},$$

where $j \geq 3$, denote a path from node B to node a_1. The source nodes m and B send 100 packets for forwarding to their respective destinations. Let us focus on node j, which performs incorrect packet forwarding for node B in steady-state with probability $B_f^{j,B}$, where $0 \leq B_f^{j,B} \leq 1$, as defined by the behavioral policy (10). Since at time t, node j formed opinion $P_{j,B}(t) \neq T_{def}$, at time t'

$$B_f^{j,B} = 1 - P_{j,B}(t).$$

Now, the key point is the opinion value that node i computes with respect to node j, after node i observes that node j performs incorrect packet forwarding for node B in steady state, with probability $B_f^{j,B}$. According to the BR-Hermes scheme, node i forms opinion $P_{i,j}(t')$ based on the information that node i collected for node j on paths R_1, R_3, R_4, since $P_{i,j}$ is a function of the counters A_j and M_j. The information that node i collected for node j on path R_1 is weighted more than the information that node i collected on paths R_3, R_4 according to (8) and (9), since node i was the source of the flow on path R_1, but an intermediate node on paths R_3, R_4. Therefore, even though node i observes that node j "punishes" node B by performing incorrect packet forwarding for node B in steady state with probability $B_f^{j,B}$ on path R_4, node i can recognize that the probability with which node j performs incorrect packet forwarding for it is $B_f^{j,i} \neq B_f^{j,B}$. This allows BR-Hermes to adopt the proposed behavioral policy. When the proposed behavioral policy is implemented by node x for node y, node x's behavior will not be misunderstood by other nodes (e.g., node z) that observe this behavior. The flow on path R_3 is considered to emphasize the fact that nodes are not assumed to have uniform packet forwarding behavior, which could be an unrealistic assumption. In this example, node j following the behavioral policy (10), performs incorrect packet forwarding for node m in steady-state with probability $B_f^{j,m} \neq B_f^{j,B}$ (where $0 \leq B_f^{j,m}, B_f^{j,B} \leq 1$).

[3] Note that in general, the nodes are mobile.

6 Security Evaluation

6.1 Attacks addressed by BR-Hermes

The BR-Hermes framework is intended to avoid a class of attacks on packet delivery in MANETs during the data transmission phase rather than the route discovery phase. The attacks on BR-Hermes are considered to be committed by "insider" nodes who have succeeded in becoming part of active routes in the network. Such nodes are owners of valid cryptographic keys or key materials and are capable of authenticating themselves as authorized users of the network. Furthermore, these nodes have successfully passed the route discovery phase of a secure routing protocol.

The integrity of message exchanges involved in the BR-Hermes protocol should be protected by cryptographic primitives such as those used in secure routing protocols. The authentication of the packets exchanged during trust establishment is discussed in the context of the E-Hermes scheme in [17]. Here, we focus our attention on insider attacks on packet forwarding and the propagation of trust information. The main attacks on packet forwarding to be considered in the attacker model include dropping, misrouting, and replaying data packets. As in secure routing protocols, sequence numbers can be used in conjunction with BR-Hermes to avoid replay attacks. The main focus of the BR-Hermes scheme lies in detecting packet dropping and misrouting attacks.

We remark that packet forwarding attacks can be launched even when a secure routing protocol is in place. A secure routing protocol aims to establish a route from a source node to a destination node containing only authorized or insider nodes. Once a route is established, nodes on the path are expected to forward packets correctly to the next hop. However, during the data transmission phase an insider node may consistently drop, misroute, or replay packets. The BR-Hermes scheme attempts to identify such misbehaviors in terms of the trustworthiness and opinion metrics, but does not purport to distinguish between malicious or non-malicious misbehaviors. Non-malicious packet forwarding misbehavior may be due to such phenomena as network congestion, node mobility, or node malfunction.

6.2 Probabilistic attacker model

The attack space covered by the BR-Hermes scheme can be described more formally in terms of a probabilistic attacker model. The probabilistic attacker model presented in this paper is an extension of the probabilistic attacker model introduced in [16]. In [16], a node was assumed to have uniform packet forwarding and trust propagating behavior towards the other network nodes. In this paper, a node may exhibit Byzantine behavior.

The attacker model consists of two types of attacks: 1) incorrect data packet forwarding; 2) incorrect propagation of trust information. Note that we do not distinguish among the various types of data packet forwarding misbehaviors, i.e., packet dropping, misrouting, and replay attacks. Incorrect trust propagation refers to a node which propagates a trustworthiness value that is different from the value that it should compute if it were following the BR-Hermes scheme. Thus, a node may propagate a trustworthiness value that is higher or lower than the value that a BR-Hermes-compliant node would compute.

Let $\mathcal{N}$ denote the set of all nodes in the network. A network attack scenario, in steady-state, is specified by characterizing, for each node $i \in \mathcal{N}$, the probability $B_f^{i,m}$ that the node performs incorrect packet forwarding for node $m \in \mathcal{N}$ and the probability $B_t^{i,m}$ that the node performs incorrect trust propagation for node m, where $0 \le B_f^{i,m}, B_t^{i,m} \le 1$. More precisely, the network attack scenario can be represented by a set of three-tuples,

$$\mathcal{S} = \{(i, B_f^{i,m}, B_t^{i,m}) : i, m \in \mathcal{N}\}. \tag{11}$$

Let η_f and η_t denote, respectively, thresholds on the degrees of packet forwarding and trust propagation misbehaviors that can be tolerated in the network. We set $\eta_f = \eta_t = T_{def}$.

The following definitions were introduced in [17] for the E-Hermes scheme and are repeated here for convenience.

Definition 1 Node i is ***good*** for node m if $B_f^i > T_{def}$.

Definition 2 Node i is defined to be ***bad*** for node m if $B_f^{i,m} < T_{def}$.

Definition 3 Node i is a ***good recommender*** for node m if $B_t^{i,m} > T_{def}$.

Definition 4 Node i is defined to be a ***bad recommender*** for node m if $B_t^{i,m} < T_{def}$.

A useful measure of the performance of the proposed trust establishment scheme is given as follows.

Definition 5 The ***bad node recognition*** percentage or *BN-recognition* is the percentage of all bad nodes that are considered bad by all of the nodes in the network.

The BR-Hermes scheme aims to identify the set of probabilities $\{B_f^{i,m}\}$ to a sufficient degree of accuracy to distinguish between good and bad nodes, based on both first-hand information from direct observations of packet forwarding behavior and second-hand information from other nodes. The probabilistic attacker model does not preclude the possibility that nodes may collude with one another. However, the BR-Hermes scheme does not seek to identify collusions per se. Rather, the BR-Hermes scheme is able to characterize the *effect* of a colluding attack as represented by an attack scenario (11).

6.3 Security properties of BR-Hermes

The attacker model presented above is simple, but sufficient to characterize the main security properties of the BR-Hermes scheme. Under BR-Hermes, the opinion metrics $P_{i,m}$ should closely approximate the underlying attack scenario under steady-state conditions. That is, in steady-state we should have

$$P_{i,m} \approx 1 - B_f^{m,i}, \quad \text{for all } i, m \in \mathcal{N}. \tag{12}$$

For the BR-Hermes framework to correctly distinguish the good nodes from the bad nodes, it is sufficient that

$$P_{i,m} > T_{def} \quad \text{for all } i \in \mathcal{N} \tag{13}$$

hold in steady-state. The simulation results presented in Section 7.3 provide validation of the steady-state properties (12) and (13).

The next few definitions were also introduced in the E-Hermes scheme [17] and are repeated here for convenience in the ensuing discussion on the security of the BR-Hermes scheme.

Definition 6 A node m is ***considered good*** by node i when the opinion $P_{i,m} > T_{def}$.

Definition 7 A node m is ***considered bad*** by node i when the opinion $P_{i,m} < T_{def}$.

Definition 8 A node m is ***considered a good recommender*** by node i when the recommender trustworthiness $T^R_{i,m} > T_{def}$.

Definition 9 A node j is ***considered a bad recommender*** by node i when the recommender trustworthiness $T^R_{i,j} < T_{def}$.

Under the probabilistic attacker model, the BR-Hermes scheme is able to distinguish the good nodes from the bad nodes in a network scenario with high accuracy, as demonstrated through the simulation results presented in Section 7.3. Nonetheless, BR-Hermes aims to calculate the opinion metrics $P_{i,m}$ accurately to closely approximate the underlying attack scenario under steady-state conditions, as characterized by (12). Note that the probabilistic attacker model only characterizes steady-state behavior. To accommodate dynamic changes in the network attack in practice, the proper use of windowing as discussed in [15,16] is necessary to maintain the responsiveness of the BR-Hermes scheme.

The key security properties provided by the BR-Hermes scheme, beyond what is provided in the previous Hermes schemes [15–17], are summarized as follows:

1. **Byzantine detection.** The proposed scheme can identify the bad nodes and bad recommenders, even when the nodes exhibit Byzantine packet forwarding and trust propagating behavior. This ability is the result of the key novel component of our scheme to weight the first-hand information that a node obtains from a traffic flow, depending on whether the given node is the source of the flow or an intermediate node on the route that the flow traverses.
2. **Byzantine robustness.** The proposed scheme is robust to the presence of bad nodes and bad recommenders, even when the nodes exhibit Byzantine packet forwarding and trust propagating behavior. Our simulation studies show very few false positives (i.e., a good node is identified as bad) and false negatives (i.e., a bad node is identified as good) even when the proportion of bad recommenders is as high as 90%. Similarly, the scheme performs well even when the proportion of bad nodes is high.
3. **Discouragement of selfish node behavior.** The proposed scheme implements a behavioral policy that discourages nodes from dropping packets to avoid being used as intermediate nodes in paths. The accuracy of the scheme is not influenced by the implementation of the behavioral policy, as demonatrated by our simulation results (see Section 7).

The BR-Hermes scheme also provides the security properties provided by the previous Hermes schemes as listed below (see [15,17]).

1. **Ability to capture independence between packet forwarding and trust propagation misbehaviors.**
2. **Resilience to the presence of bad nodes and bad recommenders.**
3. **Resilience to attacker placement.**
4. **Resilience to multiple, concurrent, and colluding attacks.**
5. **Resilience to attack frequency.**
6. **Resilience against packet duplication and replay attacks.**

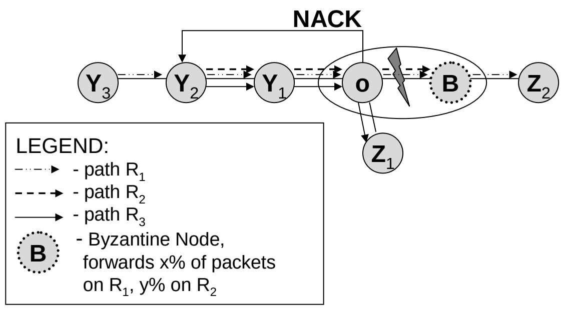


Fig. 3 Single Byzantine node.

6.4 Security analysis

We analyze the resistance of BR-Hermes to: 1) incorrect data packet forwarding and 2) incorrect propagation of trust information attacks by Byzantine nodes. The resistance of BR-Hermes to incorrect data packet forwarding and incorrect propagation of trust information attacks by non-Byzantine nodes is the same as the resistance of E-Hermes to these attacks and is discussed in [17]. Byzantine nodes perform grey-hole attacks, i.e., the Byzantine nodes drop packets of some flows that traverse them. Non-Byzantine nodes perform black hole attacks, i.e., the non-Byzantine nodes drop packets of all flows that traverse them.

Attacker nodes may impact the data flows, which traverse through them. The number of attacker nodes in the network has impact on the number of flows that may be attacked. The more the attacker nodes in the network, the more flows may be attacked. One or more attacker nodes may participate in a flow. These attackers may act independently or may collude with one another. However, the BR-Hermes scheme does not seek to identify collusions per se. Rather, the BR-Hermes scheme is able to characterize the *effect* of a colluding attack.

We shall assume that every node, whether good or bad, forwards ACK or NACK packets corresponding to packets that it has forwarded earlier. This assumption simplifies the security evaluation given below, but does not represent any limitation in the BR-Hermes scheme itself. In the BR-Hermes framework, a given node X has nothing to gain by failing to forward an ACK or NACK packet associated with a packet that it has forwarded previously. If node X fails to forward a ACK/NACK packet, node X will be penalized by all of the upstream nodes on the associated route as though it had not forwarded the original packet.

6.4.1 Byzantine nodes

Fig. 3 illustrates the response of BR-Hermes to packet forwarding misbehavior from a single Byzantine node, labeled B, on routes $R_1 = \{Y_3, Y_2, Y_1, o, B, Z_2, \cdots\}$ and $R_2 = \{Y_2, Y_1, o, B, Z_2, \cdots\}$ corresponding to flow f_1 and f_2 respectively. In particular, node B drops packets from source node Y_2 (on route R_2), but not from source node Y_3 (on route R_1). Node B's upstream neighbor, node o, observes node's B Byzantine behavior at the MAC layer (according to (8, 9)) and initiates NACKs for all packets that are not acknowledged by node B. As discussed in section 3.2, first-hand trust evaluation depends on the first-hand information gathered from a neighbor or a non-neighbor node.

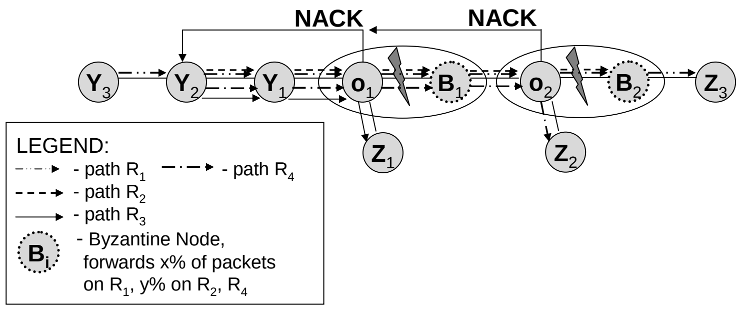


Fig. 4 Multiple non-neighbor Byzantine nodes.

– **First-hand information from neighbor node.** Node o is the FIN of Y_1 with respect to routes R_1 and R_2. Thus, node Y_1 is able to verify the correct forwarding behavior of node o, upon receiving a NACK from node o, and node Y_1 penalizes node B for the dropped packets on route R_2.
– **First-hand information from non-neighbor node.** Upon receiving a NACK initiated by node o, node Y_2 penalizes both nodes o and B. However, as a more observation data involving the two nodes with respect to different flows is accumulated over time, eventually node o will be recognized as a *good* node, whereas node X will be recognized as *bad*. For example, suppose that node Y_2 also establishes route $R_3 = \{Y_2, Y_1, o, Z_1, \cdots\}$ for its flow f_3. Node Y_2 accumulates evidence from this flow indicating that node o is a good node (for node Y_2) and identifies only node B as a bad node (for node Y_2). At the same time, nodes Y_2, Y_3 observe that node B does not drop packets of flow f_1 for source node Y_3. Nodes Y_2, Y_1, o that are upstream of node B when it exhibits Byzantine behavior, can identify node B's Byzantine behavior.

Multiple attacker nodes along a route may act independently or form collusions. Figure 4 illustrates the situation of packet forwarding misbehavior from multiple non-neighbor Byzantine nodes B_1 and B_2 on routes $R_1 = \{Y_3, Y_2, Y_1, o_1, B_1, o_2, B_2, Z_3, \cdots\}$ and $R_2 = \{Y_2, Y_1, o_1, B_1, o_2, B_2, Z_3, \cdots\}$ corresponding to flow f_1 and flow f_2 respectively. Nodes B_1 and B_2 drop a certain percentage of packets forwarded to them from source node Y_2 (along route R_2), but not from source node Y_3 (on route R_1). Here, the response of BR-Hermes is similar as in the case of a single Byzantine node depicted in Fig. 3. Note that if nodes B_1 and B_2 dropped a certain percentage of packets forwarded to them by different sources (e.g., node B_1 dropped a certain percentage of packets forwarded to it by source Y_3, whereas node B_2 dropped a certain percentage of packets forwarded to it by source Y_2), the attack scenario would be analyzed as two single Byzantine nodes on two different routes.

Node B_1's upstream neighbor node o_1 observes node B_1's Byzantine behavior at the MAC layer (according to (8), (9)) and initiate NACKs for all packets that are not acknowledged by node B_1. Similarly, node B_2's upstream neighbor node o_2, observe node B_2's Byzantine behavior at the MAC layer (according to (8), (9)) and initiate NACKs for all packets that are not acknowledged by node B_2. Node Y_1 watches on the MAC layer the forwarding behavior of node o_1 and observes that node o_1 forwards all the packets sent to it for forwarding. Thus, upon the NACK receipt from node o_1, node Y_1 knows that node B_1 is a bad node. However, node Y_1, as also node Y_2, upon a NACK receipt, which was initiated by node o_2, penalizes both nodes o_2 and B_2. Similarly, node Y_2, upon a NACK receipt, which was initiated by node o_1, penalizes both

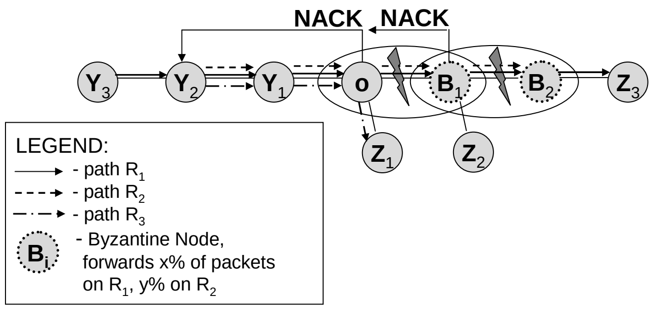


Fig. 5 Multiple neighbor Byzantine nodes.

nodes o_1 and B_1. As discussed above, nodes o_1, o_2 can be recognized as good nodes only through other flows in which nodes o_1, o_2 are not penalized.

Figure 4 summarizes the BR-Hermes response to an incorrect data packet forwarding attack of multiple neighbor Byzantine nodes on a route. This attack scenario is similar to the previous one, with the difference being that Byzantine nodes B_1 and B_2 are neighbors on routes $R_1 = \{Y_3, Y_2, Y_1, o, B_1, B_2, Z_3, \cdots\}$ and $R_2 = \{Y_2, Y_1, o, B_1, B_2, Z_3, \cdots\}$ corresponding to flow f_1 and flow f_2 respectively. Nodes B_1 and B_2 drop a certain percentage of packets forwarded to them from source node Y_2 (along route R_2), but not from source node Y_3 (on route R_1). Note that if nodes B_1 and B_2 dropped a certain percentage of packets forwarded to them by different sources, the attack scenario would be analyzed as two single Byzantine nodes on two different routes.

Node B_1's upstream neighbor node o and B_2's upstream neighbor node B_1, following the acknowledgement scheme, will initialize NACKs for all packets that are not acknowledged by nodes B_1 and B_2 respectively. Node Y_2, upon receiving a NACK initiated by node o, penalizes both nodes o and B_1. Upon receiving a NACK initiated by node B_1, node Y_2 penalizes both nodes B_1 and B_2. As discussed above, node o can be recognized as a good node only through other flows in which node o is not penalized.

6.4.2 Byzantine Recommender Nodes

The BR-Hermes scheme relies on the exchange of trustworthiness information among nodes through recommendations. Thus, an obvious attack on the BR-Hermes scheme would be for nodes to propagate false trustworthiness information. Incorrect trust propagation refers to a node which propagates an trustworthiness value that is different from the value that it should compute if it were following the BR-Hermes scheme. Thus, a node may propagate an trustworthiness value that is higher or lower than the value that a BR-Hermes-compliant node would compute. A Byzantine recommender sends different recommendations to different nodes. A node considers a recommender bad or good recommender based on the number of correct or incorrect recommendations sent to it by the recommender (see Section 6.3).

The RC-test (see Section 3.3) ensures that recommendations are accepted only when the recommended trustworthiness value is close to the first-hand trustworthiness value. If the first-hand trustworthiness value is computed from confidence smaller than c_{acc}, the node only temporarily accepts the maximum value from among all the recommenders. Because of this, bad recommender nodes are identified correctly, as also verified by our performance analysis in Section 7. Additionally, the false categorization of a recommender node as a bad recommender does not influence the correct evaluation of the nodes as good or bad. A bad recommender false positive only results in discarding the recommendations received by the recommender node, as discussed in more detail in [17].

6.4.3 Collusion of Byzantine Node and Bad Recommender

The collusion of a Byzantine node and bad recommender can be viewed as an attack on the flow, where the Byzantine node is a bad node and colludes with a bad recommender. This attack is analyzed in [17].

7 Performance Evaluation

In this section we evaluate the performance of BR-Hermes. We first discuss the convergence of the scheme and its communication and computational overhead. Then, we evaluate the accuracy of BR-Hermes by presenting some representative results from our simulation experiments.

7.1 Convergence of BR-Hermes

BR-Hermes converges to the correct value of the trustworthiness metric T, when the confidence c ($c \in [0, 1]$) associated with the trustworthiness metric T reaches one[4].

As discussed in Section 3.1, the confidence value, c, associated with the trust value t is defined in (2). The A and M parameters were introduced in Section 4.2.1. In particular, A denotes the total number of packets forwarded *correctly* (not dropped or misrouted) from the downstream nodes, whereas M denotes the total number of packets forwarded by the downstream nodes. From (2), confidence c is given by:

$$c = 1 - \sqrt{\frac{12(A/M)(1 - A/M)}{M + 1}} = 1 - \varepsilon, \tag{14}$$

where

$$\varepsilon = \sqrt{\frac{12p(1-p)}{M+1}},$$

with $p = A/M$. It is not hard to show that

$$\varepsilon \leq \sqrt{\frac{12(1/4)}{M+1}} = \sqrt{\frac{3}{M+1}} = O\left(\frac{1}{\sqrt{M}}\right). \tag{15}$$

Hence, the confidence metric converges to one at a rate of $O(1/\sqrt{M})$.

7.2 Communication and computational overhead

The extensions discussed in this paper, which enable BR-Hermes to detect Byzantine behavior and implement the proposed behavioral policy, do not impose any additional communication overhead beyond that of E-Hermes, which is discussed in [17]. The computational overhead of E-Hermes was also presented in [17]. The extensions discussed in this paper, which enable BR-Hermes to detect Byzantine behavior and implement

[4] The discussion in this section is applicable to the Hermes framework in general, but was not presented in our earlier papers.

the proposed behavioral policy, impose little additional computational overhead. The acknowledgement-related and recommendation-related computational overhead of E-Hermes is not increased by the BR-Hermes extensions.

In BR-Hermes, each node maintains a set of values associated with each of the other nodes in the network:

– Counters $\widehat{A}, \widetilde{A}, A$ and $\widehat{M}, \widetilde{M}, M$;
– Trust value $\widehat{t}, \widetilde{t}, t$ and confidence value $\widehat{c}, \widetilde{c}, c$;
– Trustworthiness value $\widehat{T}, \widetilde{T}, T$;
– Opinion value P;
– Recommender counters A^R and M^R;
– Recommender trustworthiness value T^R.

When a given node x is the source of a flow, node x updates a subset of the counters $\widehat{A}$ and $\widehat{M}$ each time it sends a packet to node y for forwarding. When node x is an intermediate node on the path that a flow traverses, node x updates a subset of counters $\widetilde{A}$ and $\widetilde{M}$ each time it sends a packet to node y for forwarding. For such a given packet, the counters $\widehat{A}, \widetilde{A}$ and $\widehat{M}, \widetilde{M}$ associated with the downstream nodes on the route are updated in accordance with the trust establishment framework discussed in Section 3.1. The values of counters A, M, trust $\widetilde{t}, \widehat{t}, t$, confidence $\widetilde{c}, \widehat{c}, c$, trustworthiness $\widetilde{T}, \widehat{T}, T$ and opinion P are then updated whenever the $\widehat{A}, \widetilde{A}$ and $\widehat{M}, \widetilde{M}$ counters are updated. The values A^R, M^R, and T^R are updated whenever the RC-test is applied (see Section 3.3). To be responsive to changes in the network dynamics, a windowing mechanism should be applied in computing the trustworthiness and opinion values [16,17]. For example, if a window of size K is used in computing the opinion value P, then the K most recent values of P over the averaging window must be stored. In summary, the storage and the computational requirements for maintaining the trustworthiness and opinion values in BR-Hermes are relatively modest.

7.3 Performance results

7.3.1 Simulation methodology

We present some representative results from our simulation experiments for evaluating the accuracy of our scheme under different network and attack scenarios. The network consists of 50 nodes that are randomly placed in a 3000 m by 3000 m area. The wireless radio transmission range of the nodes is set to 250 m. The presented results suggest the effectiveness of BR-Hermes in large network scenarios. Nodes exhibit four types of behavior.

– Type I: Good nodes and good recommenders;
– Type II: Bad nodes and good recommenders;
– Type III: Good nodes and bad recommenders;
– Type IV: Bad nodes and bad recommenders.

Randomly chosen nodes are set to exhibit Byzantine behavior, i.e. they exhibit a different type of behavior towards different nodes.

A predefined number of flows is generated for each simulation scenario. The route corresponding to a flow is not derived based on a given topology, but is chosen randomly to reflect the network topology at a given point in time. Thus, the effect of a dynamically changing network topology is captured in the simulation. The nodes in the network collect empirical evidence and build their trustworthiness and opinion values for all other network nodes based on traffic generated by the traffic flows.

Since the traffic flows are generated randomly, one or more misbehaving nodes may participate per flow. Misbehaving nodes may be neighbors or non-neighbors. We remark that in the simulations discussed here, we do not employ the averaging windows introduced in [15], in order to simplify the presentation of results. Implementation of the averaging windows would have further improved the accuracy of the final opinions when the node behaviors change over time (see Fig. 8). The presented results suggest the effectiveness of BR-Hermes in relatively large network scenarios of 50 nodes.

7.3.2 Network View without Behavioral Policy

In the first simulation scenario, the BR-Hermes scheme (with recommendations) runs without the behavioral policy being implemented. 800 flows are established along different paths in the network. The minimum and maximum number of nodes allowed on a route are five and seven respectively. Nodes $21-50$ are assigned to be of Type I for all nodes. They forward 100% of the packets that they should be forwarding and propagate correct opinion P. Nodes $1-5$ are assigned to be of Type II for all nodes and nodes $6-10$ are assigned to be of Type II for nodes $1-25$, whereas they are of Type I for nodes $26-50$. Thus, nodes $6-10$ exhibit Byzantine behavior. Nodes of Type II forward 20% of the packets received for forwarding, but propagate correct opinion P.

Nodes $16-20$ are assigned to be of Type III. Nodes of Type III forward 100% of the packets received for forwarding, but propagates recommendations of fixed opinion $P=0.5$. Nodes $11-15$ are chosen to be of Type IV for nodes $1-25$, whereas they are of Type III for nodes $26-50$. Thus, nodes $11-15$ exhibit Byzantine behavior. Nodes of Type IV forward 20% of the packets received for forwarding, and propagate recommendations of fixed opinion $P=0.5$. Although, in this case 40% of the nodes exhibit malicious behavior of one or another type, increasing this percentage does not affect the ability of the BR-Hermes scheme to form accurate opinions. The source nodes send 100 data packets during each observation window W (also called “round”). The trustworthiness parameter r is set as $r=\sqrt{2/9}$, and the RC-test threshold η (see [17]) is set to 0.1. Recommendations are exchanged among nodes that are in the same route and between any two nodes given that one of the nodes has formed opinions for nodes that the other node wants to use as intermediate nodes on a route.

Fig. 6 illustrates the opinion value that node i places on node j with a gray-scale representation. A black color implies an opinion value of 0, white represents an opinion value of 1, while intermediate values are represented by different shades of gray. Fig. 6 (a) illustrates the opinion values, $P_{i,j}$ which is the opinion formed in terms of packet forwarding. One can see that nodes $1-5$ are correctly identified as bad nodes by all other nodes (no false negatives). Nodes $16-50$ are also correctly identified as good nodes by all other nodes (no false positives). Nodes $6-15$ are correctly identified as good by nodes $26-50$ (no false positives), whereas nodes $6-15$ are in their majority correctly identified as bad nodes by nodes $1-25$ (53 false negatives). In total, 2450 opinions have been formed (50 nodes formed opinions about all other 50 nodes) and there were 53 false positives and negatives, which equals to 2.163% of false positives and negatives.

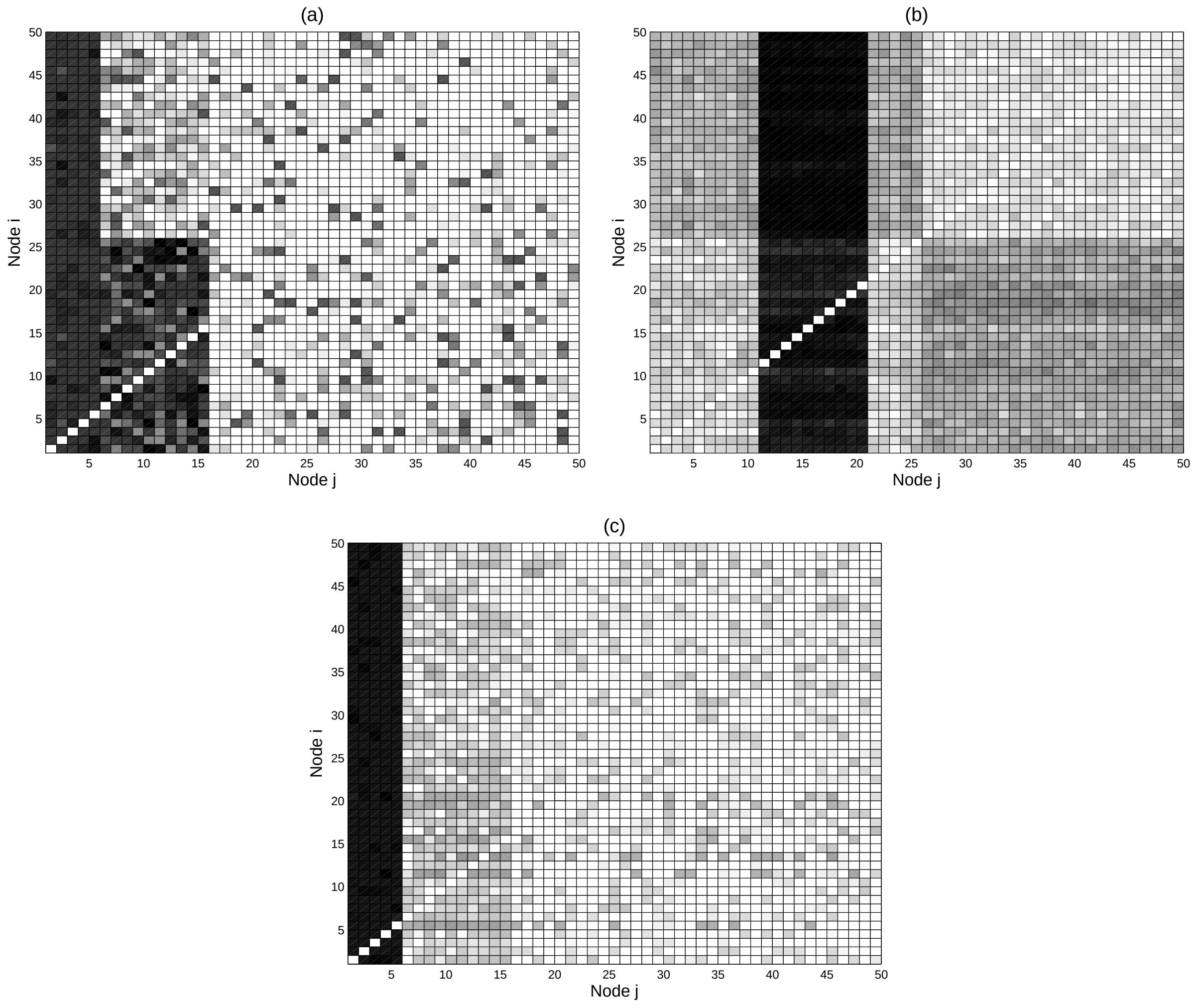


Fig. 6 Network view (a) Opinion $P_{i,j}$ for BR-Hermes, (b) Recommender trustworthiness $T^R_{i,j}$ for BR-Hermes, (c) Opinion $P_{i,j}$ for E-Hermes.

Fig. 6 (b) shows the recommender trustworthiness values, $T^R_{i,j}$, which are the opinions formed in terms of trust propagation. Nodes $11-20$ are correctly identified as bad recommenders by all other nodes. The remaining nodes are correctly identified as good recommenders. Note that nodes $26-50$ consider nodes $26-50$ as slightly better recommenders than nodes $1-25$. This is because nodes $26-50$ agree in their opinions about all other network nodes (nodes $1-50$), but do not agree with nodes $1-25$ in their opinions about all other network nodes (in particular, nodes $26-50$ do not agree with nodes $1-25$ in their opinions about nodes $6-15$). Similarly nodes $1-25$ consider nodes $1-25$ as slightly better recommenders than nodes $26-50$. This is because nodes $1-25$ agree in their opinions about all other network nodes (nodes $1-50$), but do not agree with nodes $26-50$ in their opinions about all other network nodes.

Fig. 6 (c) illustrates the opinion values, $P_{i,j}$ when E-Hermes, which does not exhibit Byzantine detection is implemented for the same simulation scenario. The bad nodes are recognized correctly, but not the Byzantine nodes. This is (as explained in details in section 4.1) because nodes do not differentiate the evidence they accumulate when they are source nodes of the flows from the evidence they accumulate when they are intermediate nodes of flows. As a result, the Byzantine nodes are viewed as good nodes always. Comparing (a) and (c), we see that BR-Hermes outperforms E-Hermes. We have also tested our scheme under various

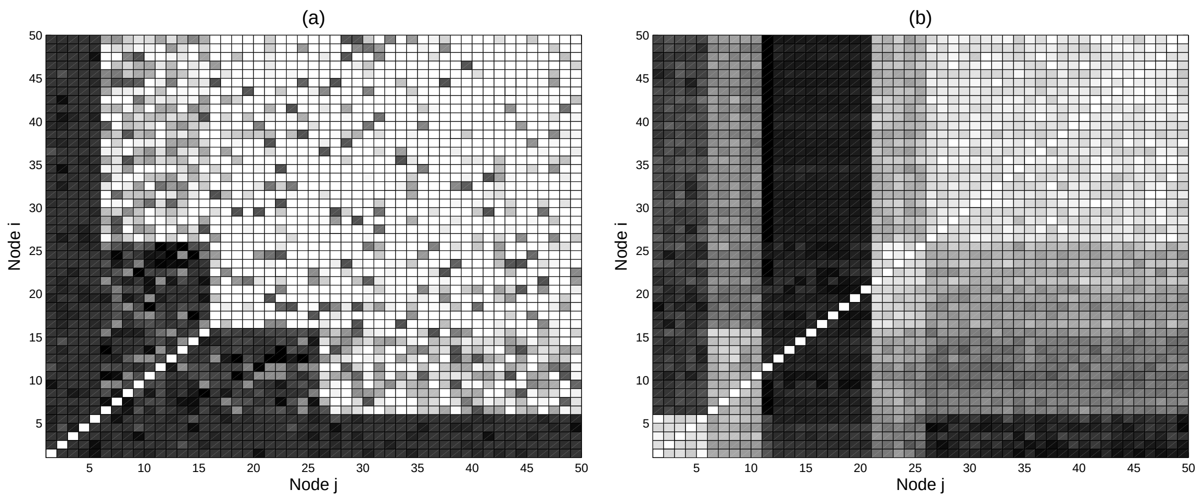


Fig. 7 Network view of BR-Hermes (a) Opinion $P_{i,j}$, (b) Recommender trustworthiness $T^R_{i,j}$.

attack scenarios, varying the number of bad recommenders and bad nodes, and found that BR-Hermes forms accurate opinions in all cases.

7.3.3 Network View of BR-Hermes

In the next simulation scenario, the BR-Hermes scheme (with recommendations) runs while the behavioral policy is implemented. The simulation scenario of Section 7.3.2 is implemented.

Fig. 7 (a) illustrates the opinion values $P_{i,j}$ with a gray-scale representation. One can see that nodes $1-5$ are correctly identified as bad nodes by all nodes (no false negatives) and that symmetrically nodes $1-5$ consider all nodes bad. This is the result of the implemented behavioral policy (10). All nodes that consider nodes $1-5$ bad, behave to nodes $1-5$ as nodes $1-5$ behave towards them, and therefore nodes $1-5$ consider them bad.

Nodes $6-15$ are correctly identified as good by nodes $26-50$ (no false positives), whereas nodes $6-15$ are in their majority correctly identified as bad nodes by nodes $1-25$ (38 false negatives). Symmetrically, nodes $6-15$ consider the majority of nodes $1-25$ bad (16 false negatives), which is the result of the implemented behavioral policy (10). All nodes that consider nodes $6-15$ bad, behave to nodes $6-15$ as nodes $6-15$ behave towards them, and therefore nodes $6-15$ consider them bad.

Nodes $16-50$ are also correctly identified as good nodes by nodes $16-50$ and nodes $6-15$ (no false positives). In total, 2450 opinions have been formed (50 nodes formed opinions about all other 50 nodes) and there were 54 false positives and negatives, which equals to 2.2% of false positives and negatives. Note that the behavioral policy does not cause false positives, due to BR-Hermes' ability to detect Byzantine behavior.

Fig. 6 (b) shows the recommender trustworthiness values, $T^R_{i,j}$, which are the opinions formed in terms of trust propagation. Nodes $11-20$ are correctly identified as bad recommenders by all other nodes. Nodes $26-50$ consider nodes $26-50$ as good recommenders and better recommenders than nodes $1-25$. In particular, nodes $26-50$ consider nodes $26-50$ as the best recommenders, nodes $21-25$ as good recommenders but slightly worse recommenders than nodes $26-50$, nodes $11-20$ bad recommenders, nodes $6-10$ as good recommenders but slightly worse recommenders than nodes $21-25$, and nodes $1-5$ worse recommenders

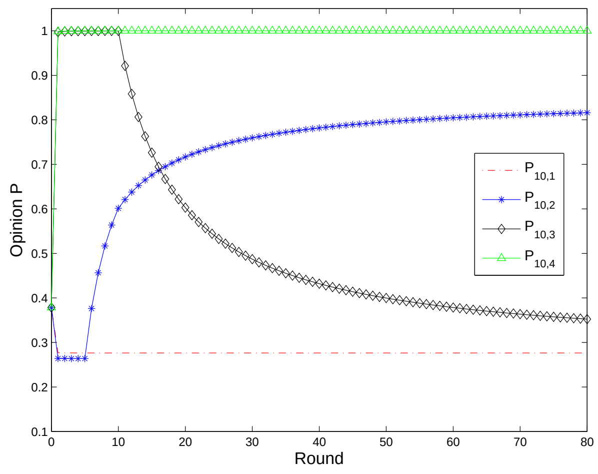


Fig. 8 Opinion that node 10 forms for nodes $1, 2, 3, 4$ from round 1 to 80. Nodes $2, 3$ change their forwarding behavior in rounds 5 and 10 respectively.

than nodes $6-10$, better recommenders than nodes $11-20$, but also as bad recommenders. This is because nodes $26-50$ agree in their opinions about all other network nodes (nodes $1-50$), but do not agree with nodes $1-25$ in their opinions about all other network nodes. In particular, nodes $26-50$ do not agree with nodes $21-25$ in their opinions about nodes $6-15$, with nodes $11-20$ in their opinions about all nodes, with nodes $6-10$ in their opinions about nodes $6-25$ and with nodes $1-5$ in their opinions about nodes $5-50$.

Nodes $16-25$ consider nodes $16-25$ as good recommenders and better recommenders than all other nodes. In particular, nodes $16-25$ consider nodes $21-25$ as the best recommenders, nodes $26-50$ as good recommenders but slightly worse recommenders than nodes $21-25$, nodes $11-20$ bad recommenders, nodes $6-10$ as good recommenders but slightly worse recommenders than nodes $26-50$, and nodes $1-5$ worse recommenders than nodes $6-10$, better recommenders than nodes $11-20$, but also as bad recommenders. This is because nodes $16-25$ agree in their opinions about all other network nodes (nodes $1-50$), but do not agree with the other nodes. In particular, nodes $16-25$ do not agree with nodes $26-50$ in their opinions about nodes $6-15$, with nodes $11-20$ in their opinions about all nodes, with nodes $6-10$ in their opinions about nodes $16-25$ and with nodes $1-5$ in their opinions about nodes $5-50$.

Similarly, nodes $6-10$ and nodes $1-5$ categorize the other nodes in 5 different groups in terms of their trustworthiness as recommenders, which can be explained by the relative opinions that they have about the network nodes. All nodes agree that nodes $11-20$ are bad recommenders.

7.4 Adaptive behavior

To demonstrate the ability of BR-Hermes, with the punishment scheme, to adapt to changes in the node behaviors, we use the same simulation scenario. A set of 100 flows are generated and the source nodes send 100 data packets during each round. The simulation runs for eighty rounds. However, now nodes $4, 6-10$ are of Type I. Nodes $3, 5$ are bad recommenders, propagating opinions with value $P = 0.5$. Nodes $1, 2$ are of Type II. Node 3 is good for rounds 1-10 and then becomes bad, thus switching from Type III to Type IV. Node 2 is bad for rounds 1-5 and then becomes good, thus switching from Type II to Type I. Node 5 is of

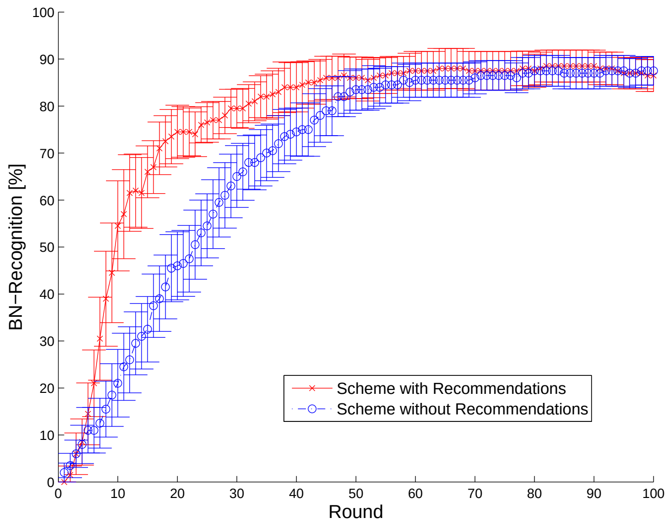


Fig. 9 Convergence comparison of scheme with and without recommendations in respect to BN-recognition.

Type III. Good nodes forward 100% of the packets that they should be forwarding. Bad nodes forward 20% of the packets received for forwarding. As before, the RC-test threshold η is set to 0.1.

The opinions P that node 10 places on nodes $1, 2, 3, 4$ over 80 rounds is shown in Fig. 8. Our scheme accurately evaluates trust and adapts to changes in the nodes' behaviors. Note that the past behavior of a node influences the value of the current opinion P. For example, at round 80 $P_{10,4} \approx 1$, whereas $P_{10,2} \approx 0.82$. Similarly, at round 80 $P_{10,1} \approx 0.27$, whereas $P_{10,2} \approx 0.35$. The implementation of the windowing mechanisms as proposed by [15] would systematically expire old observation data in order to improve the responsiveness of the system. We remark that the ability of BR-Hermes scheme to quickly adapt to changing node behavior is a key feature that makes it practical for real-world networks.

7.4.1 Convergence Comparison

In the next simulation to be presented, we compare the convergence of our scheme with and without the use of recommendations. The objective is to investigate the BN-recognition of our scheme as a function of active network flows. The simulated network consists of 10 nodes. Node 1 is a bad node. Node 3 exhibits Byzantine behavior and is bad node for nodes $1-5$ and good for nodes $6-10$. Nodes 2 also exhibits Byzantine behavior and is good for nodes $1-5$ and bad for nodes $6-10$. The rest of the nodes $4-10$ are good nodes. Bad recommenders are nodes $3, 4$ for the scheme with recommendations. As in earlier simulations, good nodes forward 100% of packets, bad nodes 20%, good recommenders propagate valid trust values, whereas bad recommenders send $P = 0.5$. Initially one flow is generated and then one flow is added per round. The flows are randomly generated. The number of nodes on a route is set to 5.

Figure 9 shows the BN-recognition of the scheme with and without recommendations. The error bars indicate the 90% confidence intervals obtained from executing on the order of 20 simulation trials for each estimated value. As expected, recommendations accelerate the convergence of the trust establishment procedures. For example, with recommendations, the BN-recognition exceeds 65% at 15 rounds, whereas when recommendations are not used the BN-recognition exceeds 65% at 32 rounds, whereas at 15 rounds it is 32%. The BN-recognition converges to a steady-state value of more than 87%.

8 Conclusion

We presented BR-Hermes, a Byzantine robust trust establishment scheme for MANETs, which is designed to improve the reliability of packet forwarding over multi-hop routes, particularly in the presence of Byzantine nodes. In the proposed scheme, each node determines the trustworthiness of the other nodes with respect to reliable packet forwarding by combining first-hand trust information obtained independently of other nodes and second-hand trust information obtained via recommendations from other nodes. BR-Hermes exploits information sharing among nodes to accelerate the convergence of trust establishment procedures, yet is robust against the propagation of false trust information by Byzantine nodes.

The proposed BR-Hermes scheme extends the E-Hermes framework introduced in [17] in several important ways. In the BR-Hermes, first-hand information that a node obtains from a traffic flow is weighted depending on whether the given node is the source or an intermediate node on the route that the flow traverses. This extension enables BR-Hermes to detect Byzantine behavior and allows us to introduce a punishment scheme that discourages selfish node behavior.

The BR-Hermes scheme allows nodes to form accurate opinions for any network node independent of whether it exhibits Byzantine behavior. The number of nodes that propagate false trust information does not influence the robustness of the system. Three types of malicious node behavior are identified: (i) dropping or misrouting packets but propagating true opinion values, (ii) forwarding packets but propagating false opinion values, and (iii) dropping or misrouting packets and propagating false opinion values. The effect of attacks by malicious nodes is identified under BR-Hermes, either when they operate separately or form collusions. Our simulation results demonstrate the effectiveness of the BR-Hermes scheme in distinguishing among malicious and non-malicious nodes in a variety of network scenarios involving Byzantine nodes that are malicious both with respect to packet forwarding and trust propagation.

Acknowledgment

This work was supported in part by the U.S. National Science Foundation under grants CCR-0209049 and CCF-0133390, and also a grant from Northrop-Grumman Mission Systems.

References

1. Avramopoulos, I., Kobayashi, H., Wang, R., Krishnamurthy, A.: Highly secure and efficient routing. In: Proc. IEEE Infocom 2004 (2004)
2. Baras, J., Jiang, T.: Cooperative Games, Phase Transition on Graphs and Distributed Trust in MANET. In: Proceedings of the 43rd IEEE Conference on Decision and Control (2004)
3. Buchegger, S., Boudec, J.Y.L.: A Robust Reputation System for P2P and Mobile Ad-hoc Networks. In: Proceedings of the 2nd Workshop on Economics of Peer-to-Peer Systems (2004)
4. Buttyan, L., Hubaux, J.P.: Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks. Mobile Networks and Applications **8**(5), 579–592 (2003)
5. Capra, L.: Engineering Human Trust in Mobile System Collaborations. In: Proceedings of the 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering, pp. 107–116 (2004)
6. Eschenauer, L., Gligor, V.D., Baras, J.: On trust establishment in mobile ad-hoc networks. In: Proc. Security Protocols Workshop, vol. 2845, pp. 47–66. LNCS (2002)

7. Hu, Y.C., Perrig, A., Johnson, D.B.: Ariadne: A secure on-demand routing protocol for ad hoc networks. In: Proceedings of the ACM MobiCom '02. ACM SIGMOBILE (2002)
8. Jiang, T., Baras, J.S.: Ant-based Adaptive Trust Evidence Distribution in MANET. In: Proceedings of the 2nd International Workshop on Mobile Distributed Computing (MDC) (2004)
9. Jiang, T., Baras, J.S.: Autonomous Trust Establishment. In: Proceedings of the 2nd International Network Optimization Conference (2005)
10. Marsh, S.: Formalizing trust as a computational concept. Ph.D. thesis, University of Stirling (1994)
11. Papadimitratos, P., Haas, Z.J.: Secure routing for mobile ad hoc networks. In: SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS) 2002 (2002)
12. Perlman, R.: Network layer protocols with byzantine robustness. Ph.D. thesis, Massachussets Institute of Technology (1988)
13. Pirzada, A.A., McDonald, C.: Establishing trust in pure ad-hoc networks. In: Proceedings of the 27th Australasian Computer Science Conference (ACSC04), pp. 47–54 (2004)
14. Theodorakopoulos, G., Baras, J.S.: Trust Evaluation in Ad-hoc Networks. In: Proceedings of the 2004 ACM workshop on Wireless Security (WiSe '04), pp. 1–10 (2004)
15. Zouridaki, C., Mark, B.L., Hejmo, M., Thomas, R.K.: A Quantitative Trust Establishment Framework for Reliable Data Packet Delivery in MANETs. In: Proc. 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05), pp. 1–10 (2005)
16. Zouridaki, C., Mark, B.L., Hejmo, M., Thomas, R.K.: Hermes: A Quantitative Trust Establishment Framework for Reliable Data Packet Delivery in MANETs. Journal of Computer Security, Special Issue on Security of Ad Hoc and Sensor Networks p. To appear (2006)
17. Zouridaki, C., Mark, B.L., Hejmo, M., Thomas, R.K.: Robust Cooperative Trust Establishment for Mobile Ad hoc Networks. In: Proc. 4rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06), pp. 23–34 (2006)