Robust Cooperative Trust Establishment for MANETs

Charikleia Zouridaki[†], Brian L. Mark[†], Marek Hejmo[†] and Roshan K. Thomas[*]

[†]Dept. of Electrical and Computer Eng.
George Mason University, MS 1G5
Fairfax, VA 22030
{czourida,bmark,mhejmo}@gmu.edu

[*]SPARTA, Inc.
5875 Trinity Parkway, Suite 300
Centreville, VA 20120
roshan.thomas@sparta.com

ABSTRACT

In a mobile ad hoc network (MANET), a source node must rely on intermediate nodes to forward its packets along multi-hop routes to the destination node. Due to the lack of infrastructure in such networks, secure and reliable packet delivery is difficult to achieve. We propose a robust cooperative trust establishment scheme to improve the reliability of packet delivery in MANETs, particularly in the presence of malicious 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. First-hand trust information for neighbor nodes is obtained via direct observations at the MAC layer whereas first-hand information for non-neighbor nodes is obtained via feedback from acknowledgements sent in response to data packets. 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. We present simulation results which demonstrate the effectiveness of the proposed scheme in a variety of scenarios involving nodes that are malicious both with respect to packet forwarding and trust propagation.

1. INTRODUCTION

In recent years, there has been considerable interest in the topic of trust establishment for ad hoc networks. Trust establishment is an important and challenging issue in the security of ad hoc networks [1]. The lack of infrastructure in a mobile ad hoc network (MANET) makes it difficult to ensure the reliability of packet delivery over multi-hop routes in the presence of malicious nodes acting as intermediate hops. In this paper, we present a robust, cooperative trust establishment scheme that enables a given node to identify other nodes in terms of how "trustworthy" they are with respect to reliable packet delivery. The proposed scheme is cooperative in that nodes exchange information in the process of computing trust metrics with respect to other nodes. On the other hand, the scheme is robust in the presence of nodes malicious nodes that propagate false trust information.

The proposed scheme extends our earlier work on *Hermes* [2], a trust establishment framework that incorporates a Bayesian approach for trust computation as well as the notion of confidence, based on first-hand observations of packet forwarding behavior obtained by neighbor nodes. In Hermes, trust establishment of non-neighbor nodes relies on the exchange of first-hand trust metrics. A drawback of the Hermes scheme is that it lacks robustness with respect to the propagation of trust information among nodes. In particular, the scheme is vulnerable to attacks by nodes that propagate erroneous trust information in the network. The trust establishment scheme proposed in the present paper avoids such attacks by extending the notion of first-hand evidence among neighbor nodes to non-neighbor nodes by employing a protocol involving acknowledgements. Thus, a given node need not rely on second-hand trust information to compute trust metrics with respect to a non-neighbor node. Nevertheless, the sharing of second-hand trust information accelerates the convergence of trust computations.

The "extended" Hermes scheme developed in this paper can be applied to any ad hoc network on top of the routing protocol. We remark that packet forwarding attacks can be launched even when a *secure* routing protocol (cf. [3, 4]) 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 Hermes trust establishment framework attempts to identify such misbehaviors in terms of trustworthiness and opinion metrics, but does not purport to distinguish between malicious or non-malicious misbehaviors. Non-malicious misbehavior may be due to such phenomena as network congestion, node mobility, or node malfunction. We note that for a trust establishment scheme to be effective, it must be capable of adapting to the dynamic changes in the topology of a MANET. This issue is addressed in the Hermes scheme [2] via the use of windowing mechanisms to systematically expire old observation data to maintain the accuracy of computed trust metrics in a dynamic network environment.

The contribution of the paper is a robust cooperative trust establishment scheme for MANETs which obtains first-hand trust information with respect to non-neighbor nodes and combines this information with second-hand trust information to accelerate the establishment of trust in an ad hoc network. The key novel components of the proposed trust establishment scheme are: (1) an acknowledgement scheme for first-hand trust information with respect to non-neighbor nodes and (2) a recommendation scheme that is robust against the propagation of false trust information by malicious nodes. The extended Hermes scheme: (i) allows nodes to form accurate opinions for any network node; (ii) models the independence of malicious behavior with respect to packet forwarding and trust propagation; and (iii) identifies the effect of attacks by individual or colluding malicious nodes. We present simulation results to demonstrate of the scheme in distinguishing between malicious vs. non-malicious nodes in a variety of scenarios involving nodes that are malicious with respect to both packet forwarding and trust propagation. In particular, we provide some numerical comparisons of the extended Hermes scheme versus the original Hermes scheme.

The remainder of the paper is organized as follows. Section 2 briefly reviews related work on trust establishment in ad hoc networks and sets the present work in context. Section 3 reviews some background material on trust establishment in MANETs. Sections 4 and 5 discuss the core concepts and advances of the paper. Section 4 discusses a protocol for accumulating trust information and computing trust metrics for non-neighbor nodes via acknowledgements. Section 5 describes a scheme for cooperatively sharing trust information among nodes via recommendations. Second-hand trust information is combined with first-hand trust information to derive an opinion metric, which summarizes the trust that a given node attributes for another node. Section 6 proposes an authentication scheme for both data packets and control packets used for trust establishment. Section 7 discusses the security properties of the trust establishment scheme. Section 8 presents results from simulation experiments that demonstrate the robustness and key properties of the proposed trust establishment scheme. Finally, the paper is concluded in Section 9.

2. RELATED WORK

In recent years, there has been considerable interest in the topic of trust establishment for ad hoc networks. The authors of [1] 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 [5] along the lines suggested in [1]. Others have approached trust establishment based on the use of a Bayesian framework [6, 2]. In this framework, a random variable that follows the beta distribution is associated with the trust value of a node. Also, the posterior distribution that represents a notion of trust is derived from a prior distribution. The Bayesian approach was initially explored in [6]. The Hermes scheme presented in [2] builds on the Bayesian approach by incorporating the notion of statistical confidence associated with a trust value.

In [7], 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 [8] and a semi-ring approach was suggested to evaluate trust and confidence along network paths. In [9], a framework for stimulating cooperation in MANETs is proposed. The approach is based on a credit system for packet forwarding while trusted hardware is assumed. The goal of collaboration is also pursued in [10], which proposes a trust management model, whereby each node carries a portfolio of credentials, which it uses to prove its trustworthiness. An autonomous trust establishment framework is proposed in [11, 12], which relies on the introduction of pre-trusted agents and a public key infrastructure.

The Hermes framework for trust management introduced in [2] 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 work in [7] uses only first-hand information, while Hermes incorporates third-party information to derive the notion of an opinion that a given node has for any other node. While many of the works deal with general notions of trust, Hermes is focused on developing metrics and mechanisms for establishing trust quantitatively with respect to the objective of reliable packet delivery.

The extended Hermes scheme proposed in the present paper addresses one of the major limitations of the original Hermes scheme in its attack model and further provides additional improvements. Hermes assumes that when a node forwards packets correctly, it also propagates trustworthiness values honestly and vice versa. However, these sets of behaviors can be independent. The focus of this paper is to extend Hermes to address an attacker model where nodes can exhibit these 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 we derive more accurate trustworthiness values for non-neighbor nodes based on first-hand information from acknowledgements, as opposed to relying only on second-hand recommendations. The use of recommendations accelerates the convergence of trust establishment in the network.

3. TRUST ESTABLISHMENT IN MANETS

In this section, we set the stage for the rest of the paper by giving a brief overview of the trust management concepts in the original Hermes scheme. For further details on the quantitative notions of trust and related trust metrics, the reader is referred to [2].

3.1 Overview of Trust Management Concepts

The notion of trust and trust relationships have been studied extensively in the literature [13]. 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* [2] 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 [2], opinions represent the combination of first-hand and second-hand i formation, the latter being gathered through recommenc tions.

3.2 Trust metrics

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

$$t \triangleq \frac{A}{A+B}, \qquad ($$

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

$$c = 1 - \sqrt{\frac{12AB}{(A+B)^2(A+B+1)}}, \qquad ($$

where $0 \leqslant c \leqslant 1$. A value of c close to one indicates hi confidence in the accuracy of the computed trust value whereas a value close to zero indicates low confidence. 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 mapping of (t, c) into a composite metric T is based on the size and shape of a family of (x,y)-ellipses as explained in [2]. The trustworthiness associated with a pair (t, c) is defined as

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

where x and y are parameters that determine 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 T_{accept} as an acceptability threshold. To meet T_{accept} the constituent confidence has to be greater than or equal to a predefined threshold ϵ. We remark that each node may choose a different value of T_{accept} to implement its own policy in determining the acceptability of trustworthiness values.

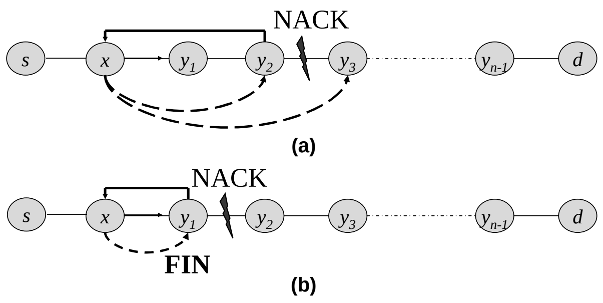

Figure 1: Processing of NACKs.

4. FIRST-HAND TRUST EVALUATION

In this section, we present a new scheme for gathering first-hand trust information from non-neighbor nodes. This is an extension over Hermes, which gathers first-hand information only from neighbor nodes. The scheme requires authentication mechanisms for both data and control packets.

4.1 Neighbor nodes

We first review the Hermes approach for establishing trust for neighbor nodes introduced in [2]. In the Hermes scheme, nodes evaluate the trustworthiness of their neighbors by snooping the wireless channel. Omnidirectional antennas and the absence of dynamic power control are assumed.

Consider a very simple route $\{x, y, z\}$. In this scheme, a given node x in the network maintains counters M_y and A_y for a neighbor node such as 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.

The counters M_y and A_y are updated as follows. Whenever a packet p is forwarded from node x to node y, M_y is incremented by one and a timer is initiated. The timeout interval is set to a value greater than the maximum round-trip time (RTT) between two neighbor nodes in the network. If node x observes a copy of packet p forwarded from node y correctly to the next hop (say node z) before the timer expires the counter A_y is incremented by one. Otherwise, the counter A_y is not updated.

4.2 Non-neighbor nodes

We now generalize the evaluation of first-hand trust to non-neighbor nodes. We shall assume that source routing [14, 15] is used, so that the complete path taken by the packet is known. Source routing protocols tend to be easier secure than other types of routing protocols, e.g., distance vector-based routing protocols.

We gather first-hand information from non-neighbor nodes through the use of an acknowledgements scheme. Consider the topology given in Fig. 1. When node x forwards packet p to node y_1, it initiates an *acknowledgement timer* with timeout interval t^{ack} and updates the M-counters for the downstream intermediate nodes as follows:

$$M_{y_i} \leftarrow M_{y_i} + 1, \quad 1 \leq i \leq n-1. \tag{4}$$

The value of timeout interval t^{ack} should be larger than the maximum round-trip propagation time along the given path in the network.

In the case of an acknowledgement (ACK) packet, node x forwards the ACK to its upstream neighbor in general (either another intermediate node or the source node itself), and updates the A-counters for all of the downstream intermediate nodes as follows:

$$A_{y_i} \leftarrow A_{y_i} + 1, \quad 2 \le i \le n-1, \tag{5}$$

which indicates that all of the downstream nodes had correctly forwarded the packet p. Since node y_1 is a direct neighbor of node x, the counter A_{y_1} is updated based on observation of the packet forwarding behavior at the MAC layer as discussed earlier in Section 4.1.

Let us consider the case when an ACK is not received. There are three subcases. First, if an intermediate node along the path R_p fails to receive an ACK or negative acknowledgement (NACK) packet within the timeout period, it creates its own NACK packet and then sends it to the upstream neighbor on the path. The other two subcases are illustrated in Fig. 1 (a dashed arrow from node x to node y indicates that x assumes y to be malicious or faulty and a regular arrow indicates the transmission of a NACK). Figure 1(a) illustrates the subcase where node x receives a NACK from node y_2 and Fig. 1(b) illustrates the subcase when node x receives a NACK from node y_1, which we call the *first intermediate node* (FIN). We now discuss the two subcases separately:

1. *NACK originating from node* y_i, $2 \le i \le n-1$: In this case, node x infers that a fault occurred on the link (y_i, y_{i+1}), but cannot identify which of the two nodes y_i and y_{i+1} caused the fault (y_2 or y_3 in Fig. 1(a)). Therefore, node x suspects both nodes as faulty. To avoid unnecessarily penalizing the nodes downstream from y_{i+1}, the M-counters for these nodes are decremented by one as follows:

$$M_{y_j} \leftarrow M_{y_j} - 1, \quad i+2 \le j \le n-1. \tag{6}$$

On the other hand, the intermediate nodes $y_1, \cdots, y_{i-1}$ should receive credit for correctly forwarding the packet. This is done by incrementing the corresponding A-counters by one:

$$A_{y_j} \leftarrow A_{y_j} - 1, \quad 1 \le j \le i-1. \tag{7}$$

2. *NACK originating from FIN node* y_1: If node x had previously observed at the MAC layer that node y_1 correctly forwarded packet p, then node x assumes that node y_2 failed to forward the packet correctly. In other words, the FIN property of y_1 implies that x can monitor the forwarding behavior of y_1 at the MAC layer and this allows it to narrow the fault to y_2. To avoid penalizing the nodes downstream from y_2, the M-counters for the nodes $y_3, \cdots, y_{n-1}$ are decremented by one:

$$M_{y_i} \leftarrow M_{y_i} - 1, \quad 3 \le i \le n-1. \tag{8}$$

On the other hand, if node x had observed at the MAC layer that node y_1 incorrectly forwarded the packet p (see Fig. 1(b)), then the nodes downstream from y_1 should not be penalized. Therefore, node x decrements the M-counters for these nodes by one:

$$M_{y_i} \leftarrow M_{y_i} - 1, \quad 2 \le i \le n-1. \tag{9}$$

Note that the counters are maintained only for downstream nodes. The reason is that 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 nodes of a 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.

4.3 Computing Trustworthiness

Given the counters M_y and A_y, maintained for both neighbor and non-neighbor nodes with which a source node interacts, the number of packets forwarded *incorrectly* by node y is given by $B_y \triangleq M_y - A_y$. Then the trust and confidence that x attributes to y over an observation window are given by (cf. (1) and (2))

$$t_y = t(A_y, B_y) \text{ and } c_y = c(A_y, B_y),$$

from which the trustworthiness value T_y can be computed via (3). In the next section we discuss how trustworthiness calculations from neighbor and non-neighbor nodes are combined to formulate *opinions*.

5. FORMULATION OF OPINIONS

Node i may need to make routing or other network-related decisions that involve nodes, for example, a node m for which trustworthiness value $T_{i,m}$ is below T_{accept}. In this case, second-hand trustworthiness values from third-party nodes are incorporated to form an opinion about node m. The propagation of trustworthiness information to form opinions is accomplished through *recommendations*.

5.1 Recommendations and Related Definitions

DEFINITION 1. *A* **recommendation** *by node j on node m is an assertion by j of the trustworthiness, which it has for node m (denoted as $T_{j,m}$). Node j is thus the* **recommender**.

Node i seeks recommendations on a node m when the trustworthiness it has computed for m is below T_{accept}. Node i discriminates among multiple recommenders by evaluating a metric called *recommender trustworthiness*.

DEFINITION 2. **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.*

DEFINITION 3. *A node j is considered a* **good recommender** *by node i when the recommender trustworthiness $T^R_{i,j}$ that i places on recommender j exceeds T_{def}.*

DEFINITION 4. *A node j is considered a* **bad recommender** *by node i when the recommender trustworthiness $T^R_{i,j}$ that i places on recommender j is smaller than T_{def}.*

DEFINITION 5. *A node j is considered* **good** *by node i when the trustworthiness $T_{i,j}$ that i places on j exceeds T_{def}.*

DEFINITION 6. *A node j is considered* **bad** *by node i when the trustworthiness $T_{i,j}$ that i places on j is smaller than T_{def}.*

Definition 7. *The* **bad node recognition** *percentage or* **BN-recognition** *is the percentage of all bad nodes that are recognized as bad by all of the nodes in the network.*

5.2 Processing recommendations

Consider a scenario where node i asks a set of nodes D for their recommendations for node m. Recommendations are sought when a node wishes to establish a route in which some of the nodes have a trustworthiness value is smaller than T_{accept}. The *recommender set* D is chosen from among all nodes in the network in the following order of priority: (i) good recommenders, (ii) nodes for which the recommender trustworthiness $T^R > T_{accept}$, and (iii) all other bad recommenders. We remark that bad recommenders may be chosen as part of the recommender set in order to update their recommender trustworthiness values. The recommender set D is limited to a size d to limit the communication overhead. No mechanisms are in place to obligate nodes to respond to recommendation requests. We assume that node i will receive $f \leq d$ recommendations due to network conditions or lack of willingness to respond to the request. Additionally, when node j places on node m trustworthiness smaller than T_{accept}, j does not reply to node i's recommendation request. Recommendations are authenticated with a message authentication code (MAC) computed using the shared keys between the source s and the destination d of the request or the reply.

Suppose node i receives recommendations for node m. For a given recommendation received from node $j \in D$, the procedure below is followed. If node i has formed for node m a trustworthiness value smaller than T_{accept}, it temporarily accepts the maximum value from among all the recommenders. This temporary trustworthiness value is used for routing or any other network-related decisions until subsequent updates lead to node i's trustworthiness exceeding T_{accept}. Then (i) a test is run to determine the trustworthiness of the recommendations, (ii) the recommender trustworthiness is updated and (iii) node i forms its opinion $P_{i,m}$ for node m. The procedures of the last three steps are discussed in the following two subsections. Algorithm 1 summarizes the procedure that node i executes to process recommendations for a node m.

Algorithm 1 Processing of recommendations for node m

```
choose recommender set $D$
obtain recommendations for nodes in $D$
if $T_{i,m} < T_{accept}$ then
    $T_{i,m}^{tmp} \leftarrow \max\{T_{j,m} : j \in D\}$
end if
run RC-test for recommendations $T_{j,m}, \forall j \in D$
update recommender trustworthiness $T_{i,j}^R, \forall j \in D$
calculate opinion $P_{i,m}$
```

5.3 Recommender trustworthiness

When $T_{i,m} > T_{accept}$, the trustworthiness of the recommenders $j \in D$ can be evaluated. This is done by performing the following *recommender's test* or *RC-test*:

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

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 B, where A counts the number of times for which the RC-test succeeds and B counts the number of times for which the RC-test fails. The A and B counters are then used to calculate the *recommender trustworthiness* $T_{i,j}^R$ according to the trustworthiness formulas (1)- (3).

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. As discussed in section 4.2, when node m sends a NACK upstream, the source node i suspects both m and its adjacent downstream node as faulty. On the other hand, since j is a neighbor of m, it can isolate the fault either to node m or its downstream neighbor. In this case, the trustworthiness that i calculates m, $T_{i,m}$, and the trustworthiness that j calculates for m, $T_{j,m}$, could be significantly different when m is actually a good node. Thus, the RC-test would fail for node j even though it may in fact be a good recommender.

5.4 Definition of Opinion

We now generalize the notion of trustworthiness to the concept of *opinion*, which incorporates second-hand trustworthiness values from third-party nodes. We denote the opinion that node i has for node m by $P_{i,m}$. We now provide the definition for the opinion that any node i has for another node m as follows:

$$P_{i,m} \triangleq \max_{j\in\Gamma}\{\omega_{i,j}T_{j,m}\}, \text{ for } P_{j,m} \neq T_{def} \tag{10}$$

$$\omega_{i,j} = \begin{cases} T_{i,j}^R, & i \neq j, \\ 1, & i = j. \end{cases} \tag{11}$$

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

6. AUTHENTICATION OF PACKETS

Authentication of every data, ACK, and NACK packet is required to protect the network against modification and impersonation attacks. In the extended Hermes scheme, we require all nodes to verify the authenticity of ACK/NACK packets received from downstream nodes in order to draw reliable conclusions about their forwarding behaviors. Hash chains provide a convenient mechanism for authentication to be performed by all upstream nodes. In this section, we propose an extension of the mechanisms proposed in [16, 17] to provide authentication of packets in the extended Hermes scheme. We remark when a secure routing algorithm is in place, 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.

6.1 Data packets

As in [16], the authentication field of a data packet consists of a sequence of message authentication codes (MACs) computed using the shared keys between a source s and each of the intermediate nodes a_i and the destination d. The MAC of node a_i receives as input the data packet and the MACs for node $a_{i+1}, a_{i+2}, ..., d$. This way, the data packets enable hop-by-hop authentication verification that protects against malicious intermediate nodes trying to tamper with the MAC field of a downstream node without being detected.

6.2 Control packets

As in [16], the authentication fields of an ACK/NACK packet satisfy two properties; (i) they are impractical to forge and (ii) if an ACK/NACK verifies at one non-faulty node on the path, it satisfies at all non-faulty routers on the path. A one-way hash function $h(.)$ and hash chains of length three are used to guarantee the previous properties.

Consider a path $R = \{s, a_1, a_2, \cdots, a_{n-1}, a_n = d\}$, where $n \geq 2$, from s to d and denote the sequence number of a packet p as k and the shared key between source s and node a_i as $K_s^{a_i}$. Each source s constructs $n-1$ hash chains to be used when the ACK of p is forwarded and $n-2$ hash chains (no hash chain for destination d) to be used when the NACK of p is forwarded. As we shall explain shortly, a total of $2n-3$ hash chains need to be initiated by the source to ensure that malicious intermediate nodes cannot discard the MAC field of another node without being detected.

Similar to the notation used in [16], we denote by $r_i^0(k|0)$ the first element of the "ACK" hash chain for node a_i; its second element is used to authenticate an ACK. The hash chain element $r_i^0(k|0)$ is constructed by concatenating the key $K_s^{a_i}$, the sequence number k and the element 0. Similarly, we denote by $r_i^0(k|1)$ the first element of the "NACK" hash chain for node a_i; its second element will be used to authenticate a NACK. The element $r_i^0(k|1)$ is constructed by concatenating the key $K_s^{a_i}$, the sequence number k, and the element 1. The succeeding elements $r_i^l(k|0)$, $r_i^l(k|1)$, $l = 1, 2$, are constructed by applying a one-way hash function $h(.)$ to the immediately preceding elements.

The source node announces, using the data packet, all $2n-3$ third elements, which are protected by the authentication field of the packet. Each recipient a_i is able to construct $r_i^0(k|0)$, $r_i^0(k|1)$. When destination d sends an ACK, it appends $r_n^1(k|0)$ to the ACK packet. The element $r_n^1(k|0)$ allows node $a_i$and s to identify that d send the ACK. Similarly, every a_i appends $r_i^1(k|0)$ and the ACK reaches the source s. If a timeout occurs at node a_i, a_i constructs a NACK and appends element $r_i^1(k|1)$ to authenticate it. Similarly, all the upstream nodes of a_i append their authenticators to the NACK.

In [16], only $n-1$ hash chains, corresponding to the "ACK" hash chains discussed above, are constructed by the source node i. However, this scheme is vulnerable to the following attack. Consider the path $R = \{s, a_1, a_2, a_3, a_4 = d\}$. Suppose that destination d sends an ACK, and appends the element $r_4^1(k)$. Nodes a_3 and a_2 properly append $r_3^1(k)$ and $r_2^1(k)$ respectively to the ACK packet received and forward it to node a_1. If a_1 is a malicious node, it can discard the authenticators $r_4^1(k)$ and $r_3^1(k)$, append $r_1^1(k)$ and thus forward

$$(ACK|r_2^1(k)|r_1^1(k))$$

to the source s. The source will then believe that node a_2 constructed a NACK for link (a_2, a_3) and nodes a_2, a_3 will be erroneously penalized as being faulty.

In the proposed scheme with both "ACK" and "NACK" hash chains, the malicious node a_1 cannot discard the authenticators $r_4^1(k|0)$ and $r_3^1(k|0)$ and append $r_1^1(k|1)$, without being detected. The reason is that

$$(NACK|r_2^1(k|0)|r_1^1(k|1))$$

would be forwarded to the source s, which would see the inconsistency of the received packet ($r_2^1(k|0)$ authenticates an ACK, whereas $r_1^1(k|1)$ authenticates a NACK) and correctly identify link (a_1, a_2) as faulty. Thus, $2n-3$ hash chains are required because authenticators $r_i^1(k)$ authenticates only the node that appended the authenticator, not the packet content. Note that hop-by-hop authentication of the ACK and NACK packets is required for recognition of the faulty link, which allows the source and intermediate nodes to collect valid information about the forwarding behavior of all their downstream nodes.

7. SECURITY PROPERTIES

In this section we describe informally some of the key security properties that our scheme guarantees. Simulations results that illustrate some of these properties are discussed in the next section.

1. **Ability to model independence in malicious behaviors.** The proposed scheme can handle the dropping and misrouting of packets, as well as the propagation of false opinion values. Even when false trustworthiness values are computed for recommenders, the scheme will converge to the correct opinions reflecting the underlying packet forwarding behavior.

2. **Robustness against false recommendations.** Our scheme is robust against scenarios where the majority of the nodes send false recommendations. Our simulation study shows very few false positives (a good node is identified as bad) even when the proportion of bad recommenders is 90%. Our scheme is also resilient against the presence of bad nodes. In summary, neither the number of bad recommenders nor bad nodes compromises the ability of the scheme to form correct opinions.

3. **Ability to distinguish the bad node among two neighbors.** The ACK processing scheme may identify two neighbor nodes as malicious with respect to a given flow, even when one of the nodes is good. However, if there are multiple flows on different routes to which these nodes belong, the scheme will be able to isolate the bad node among the two.

4. **Resilience against multiple, concurrent and colluding attacks.** Three properties in the scheme collectively ensure resilience against multiple and concurrent as well as colluding attacks: (i) the ability of a sender to overhear and verify forwarding of its packets by its FIN nodes, (ii) an acknowledgement scheme that conservatively identifies potential malicious nodes, and (iii) a node i evaluates recommendations from another node j on node m always in relation to the first-hand trustworthiness that node i has formed on m.

5. **Robustness to attacker placement and attack frequency.** The accuracy of our scheme does not depend on the frequency with which an attack is performed or the placement of the attackers along a path. This is because a sender node can always monitor the

FIN nodes of flows. Further, the monitoring and processing of packet and acknowledgement forwarding are independent of the attack frequencies.

6. **Resilience against duplication and replay.** In the Bayesian framework statistical confidence increases as the number of data samples increases. As such, it may appear at first sight that our scheme can be manipulated to boost confidence through packet duplication and replay. We mitigate these attacks through the use of sequence numbers at the network layer. Attacks associated with data packet replays are stopped at the FIN, which disregards a data packet with a replayed sequence number. As a result, the remaining downstream nodes do not receive packet duplicates.

8. PERFORMANCE EVALUATION

In this section we evaluate the performance of Hermes. We first evaluate the accuracy of Hermes by presenting some representative results from our simulation experiments.

8.1 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 10 nodes that are randomly placed in a 500 m by 500 m area. The wireless radio transmission range of the nodes is set to 250 m. Nodes exhibit four types of behavior.

- Type I: Good node, good recommender;
- Type II: Bad node, good recommender;
- Type III: Good node, bad recommender;
- Type IV: Bad node, bad recommender.

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. The number of the traffic flows generated in the simulation scenarios presented in this section is chosen to be small to highlight the convergence of our scheme. However, when the number of generated flows is small, some nodes may not participate in any flows and as a result, no opinion is formed for them. Given a sufficiently large set of traffic flows, all nodes should be able to form valid opinions for every other node in the network. We remark that in the simulations discussed here, we do not employ the averaging windows introduced in [2], 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. 4).

8.2 Network View

In the first simulation scenario, eight random traffic flows are established along different paths in the network. The minimum and maximum number of nodes allowed on a route are four and seven respectively. Nodes $1, 3, 4, 5, 8, 9, 10$ are randomly assigned to be of Type I. They forward 100% of the packets that they should be forwarding and propagate correct opinions P. Node 7 is randomly assigned to be of Type II. Node 7 forwards 20% of the packets received for forwarding, but propagates correct opinions P. Node 6 is randomly assigned to be of Type III. Node 6 forwards 100% of the packets received for forwarding, but propagates recommendations of fixed opinion $P = 0.5$. Node 2 is randomly chosen to be of Type IV. Node 2 forwards 20% of the packets received for forwarding, and propagates recommendations of fixed opinion $P = 0.5$. Although, in this case 30% of the nodes exhibit malicious behavior of one or another type, increasing this percentage does not affect the ability of the extended Hermes scheme to form accurate opinions. The source nodes send 100 data packets during each observation window W (also called "round"). The trustworthiness parameters are set as $x = \sqrt{2}$ and $y = \sqrt{9}$, and the RC-test threshold η is set to 0.1.

First, we implement our scheme with recommendations. 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. Figure 2 shows the opinion P and trustworthiness T^R that good node (and good recommender) 10 places on all other network nodes after 1, 3, 10, and 30 rounds. We note that node 10 identifies correctly the type of behavior of each node. Type I nodes appear in the top-right corner area, Type II nodes appear in the lower-right corner area, Type III nodes appear in the upper-left corner area, whereas Type IV nodes appear in the lower-left corner area. Node 3 is correctly identified as good node. However, node 10 has not formed recommender trustworthiness T^R for it, $T^R_{10,3} = T_{def}$, as node 10 had not asked node 3 for recommendations. Observe that the more observations node 10 makes, the more accurately it assigns opinion and recommender trustworthiness values.

Figure 3 illustrates the opinion value that node i places on node j in a grayscale representation. The color black represents an opinion value of 0, white represents an opinion value of 1, while intermediate values are represented by different shades of gray. Figure 3 (b) illustrates the opinion values, $P_{i,j}$ which is the opinion formed in terms of packet forwarding. One can see that nodes 2 and 7 are identified as bad nodes by all other nodes, but node 6, which has not yet formed an opinion about these nodes. Node 7 has not identified node 2 as bad node for the same reason. The good nodes are also correctly identified.

Figure 3 (c) shows the recommender trustworthiness values, $T^R_{i,j}$, which are the opinions formed in terms of trust propagation. Nodes 2 and 6 are correctly identified as bad recommenders by all other nodes that were able to form acceptable recommender trustworthiness values T^R for them. The remaining nodes are correctly identified as good recommenders with one exception. There is a false positive recommender trustworthiness value T^R, due to the fact that only eight flows are active. With more flows in the network, the accuracy of the opinion values formed improves. However, note that the existence of false positives T^R is acceptable,

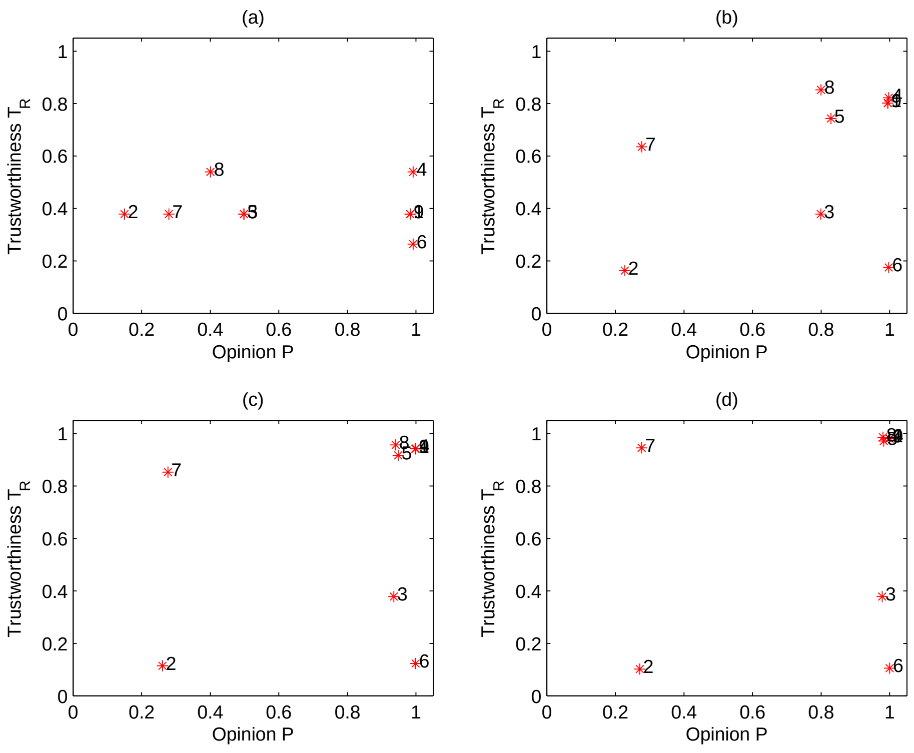

Figure 2: Opinion of good node/recommender for all other network nodes after (a) 1 round, (b) 3 rounds, (c) 10 rounds, (d) 30 rounds.

as long as the correct opinion values are formed, which is the case here.

Figure 3 (a) illustrates the opinion values, $P_{i,j}$ when the scheme without recommendations is implemented for the same simulation scenario. Nodes that have interacted with nodes 2 and 7 have correctly identified them as bad nodes. Nodes that interacted with the remaining nodes have identified them as good, with two exceptions. Two false positives are attributed to the fact that upon receipt of a NACK, both nodes of the faulty link are assumed to be bad. This effect can be attenuated with the presence of a larger number of distinct active flows containing both nodes. Comparing (a) and (b) we see that when recommendations are used, nodes form the correct network view much more quickly. We have also tested our scheme under various attack scenarios, varying the number of bad recommenders and bad nodes, and found that the scheme forms accurate opinions in all cases.

8.3 Adaptive behavior

To demonstrate our scheme's ability (with recommendations) to adapt to changes in the node behaviors, we use the same simulation scenario. Eight flows are generated and the source nodes send 100 data packets during each round. The simulation runs for fifty rounds. However, now nodes $1, 4, 5, 8, 9, 10$ are of Type I. Nodes $2, 6$ are bad recommenders, propagating opinions with value $P = 0.5$. Node 3 is of Type II. Node 2 is good for rounds 1-5 and then becomes bad, thus switching from Type III to Type IV. Node 7 is bad for rounds 1-10 and then becomes good, thus switching from Type II to Type I. Node 6 is of 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 $2, 3, 7, 8$ over 50 rounds is shown in Fig. 4. 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 50 $P_{10,8} \approx 1$, whereas $P_{10,7} = 0.86$. The implementation of the windowing mechanisms as proposed by [2] would systematically expire old observation data in order to improve the responsiveness of the system. We remark that the ability of Hermes scheme to quickly adapt to changing node behavior is a key point of the scheme that makes it practical for real-world networks.

8.4 Convergence comparison

In the next simulation scenario, we compare the convergence of our scheme with and without the use of recommendations. The objective is to investigate the BN-recognition (see Section 5.1) of our scheme as a function of active network flows. The simulated network consists of 10 nodes. Nodes $1, 3, 4, 5, 8, 9, 10$ are of Type I, node 7 is of Type II, node 6 is of Type III and node 2 of Type IV. 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 5 shows the BN-recognition of the scheme with and without recommendations. The error bars indicate the 90% confidence intervals obtained from executing on the

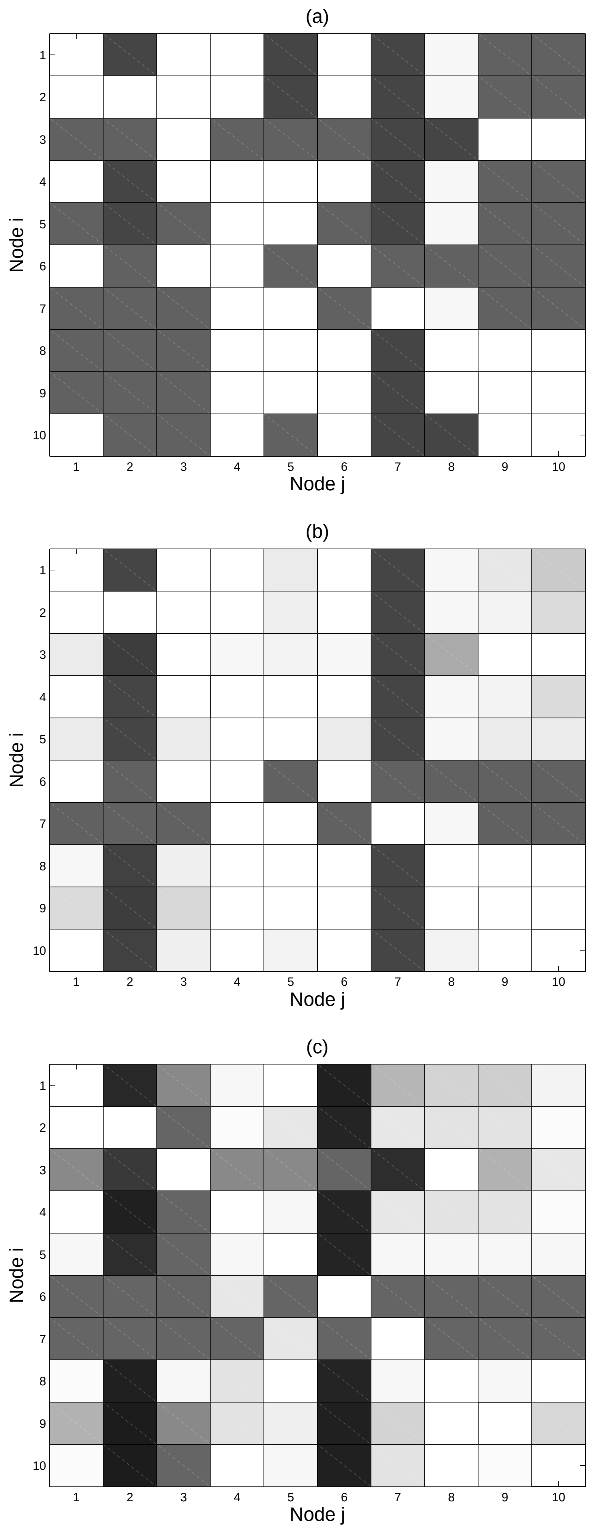

Figure 3: Network view: (a) Opinion $P_{i,j}$, without recommendations, (b) Opinion $P_{i,j}$, with recommendations, (c) Recommender trustworthiness $T^R_{i,j}$.

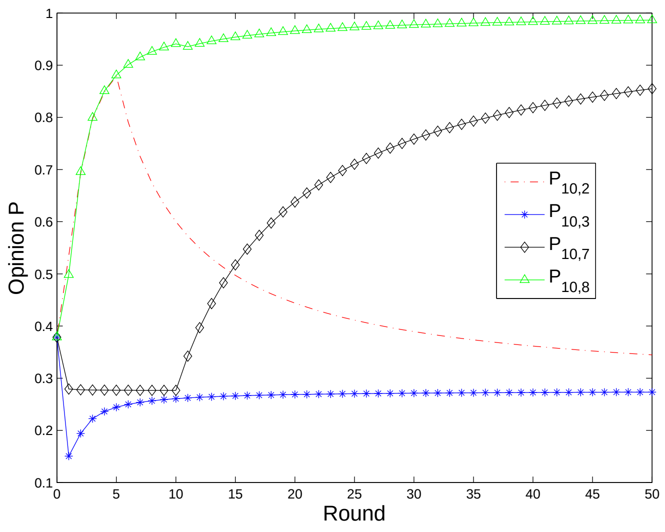

Figure 4: Opinion that node 10 forms for nodes 2, 3, 7, 8 from round 1 to 50. Nodes 2, 7 change their forwarding behavior in rounds 5 and 10, respectively.

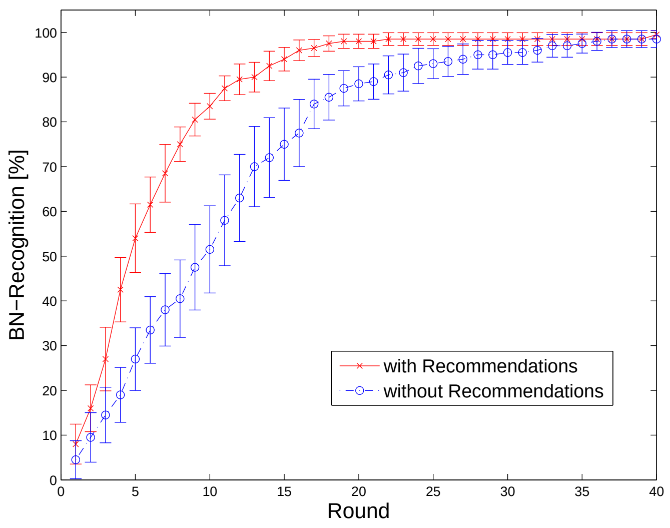

Figure 5: Convergence comparison of scheme with and without recommendations in respect to BN-recognition.

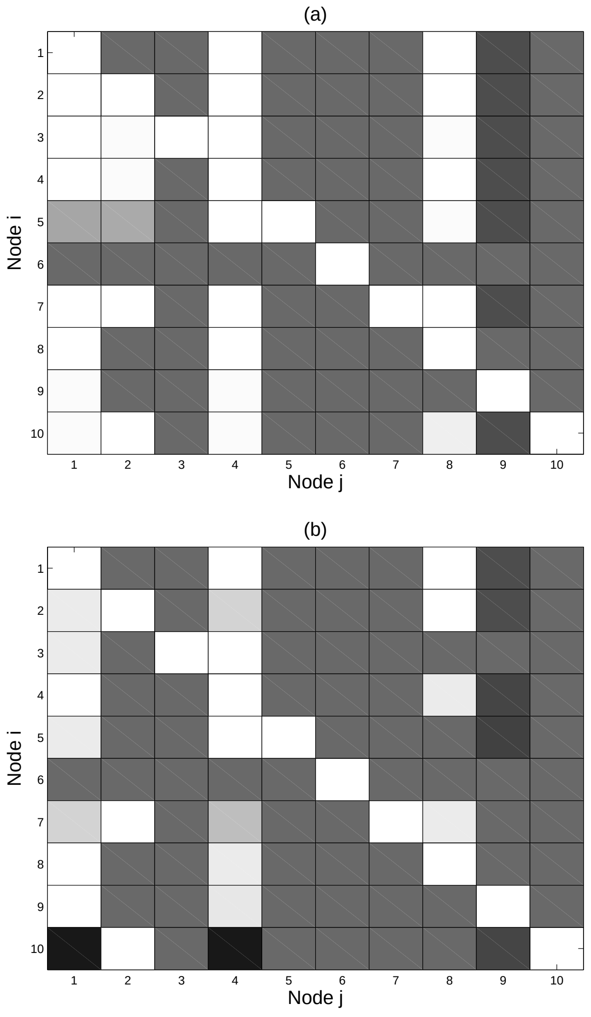

Figure 6: Network view: (a) Opinion $P_{i,j}$ of extended Hermes, (b) Opinion $P_{i,j}$ of Hermes.

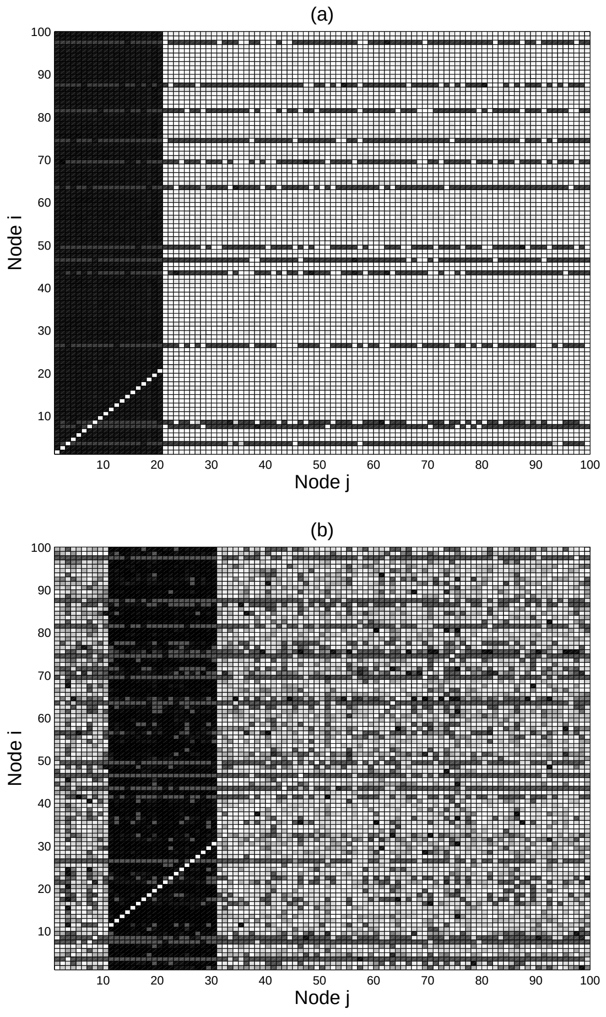

Figure 7: Large-scale network view: (a) Opinions $P_{i,j}$, (b) Recommender trustworthiness values $T^R_{i,j}$.

order of 20 simulation trials for each estimated value. As expected, recommendations accelerate the convergence of the trust establishment procedures. With recommendations, the BN-recognition converges to a steady-state value after fewer than 20 rounds, whereas when recommendations are not used, more than 35 rounds are required for the scheme to converge.

8.5 Extended Hermes vs. original Hermes

Figure 6 compares the performance of the extended Hermes scheme with the original Hermes scheme proposed in [2] in a gray-scale representation. The comparison is done for the simulation scenario presented in [2]. Five traffic flows are established in a network of ten nodes. Node 9 acts maliciously, forwarding only 20% of the packets. Hermes assumes that malicious nodes are also bad recommenders. The trust values they propagate are ignored and T_{def} is used for the trustworthiness calculation of nodes downstream of a malicious node. We simulated a scenario in which node 9 is a bad recommender that propagates $P = T_{def}$. All other nodes forward 90% of the packets they should be forwarding. The source node of each flow sends 20 packets per window over the course of 30 rounds.

Figure 6 (a) illustrates the opinion values $P_{i,j}$, that node i places on node j when our scheme is implemented and Fig. 6 (b) is the equivalent network view presented in [2]. One can verify that the extended Hermes scheme is superior in terms of (i) convergence time and (ii) accuracy of the opinion values in the sense that more nodes correctly identify other nodes as good or bad. In Fig. 6 (a), we see that seven nodes have identified node 9 as bad, whereas in Fig. 6 (b) only five nodes have identified node 9 as bad.

8.6 Large network scenario

We now simulate a larger network scenario consisting of 100 nodes and 200 random traffic flows established along different paths in the network. The minimum and maximum number of nodes allowed on a route are four and ten, respectively. We chose 2 flows to be initiated in average per node to emphasize the convergence of our scheme with a limited number of traffic flows. Nodes 1-10 are assigned to be of Type II and forward 20% of the packets received for forwarding, but propagate correct opinions P. Nodes 21-30 are assigned to be of Type III; they forward 100%

of the packets received for forwarding, but propagate recommendations of fixed opinion $P = 0.5$. Nodes 11-20 are chosen to be of Type IV; they forward 20% of the packets received for forwarding, and propagate recommendations of fixed opinion $P = 0.5$. The remaining nodes are of Type I; they forward 100% of the packets that they should be forwarding and propagate correct opinions P. Thus, 30% of the network nodes exhibit malicious behavior of one or another type. The source nodes send 50 data packets during each observation window W. The trustworthiness parameters are set as as before. Recommendations are used in the simulation.

Figure 7 (a) illustrates the opinion values $P_{i,j}$, that node i places on node j. One can see that nodes 1-20 are identified as bad nodes and nodes 21-100 are identified as good nodes by 87% of nodes; 13% nodes did not form an opinion about one or more of nodes, because of lack of interaction with them. In total, $9,900$ opinions are formed. The number false positives was 16, which correspondes to 0.0016% of all opinions. The false positives are attributed to the fact that upon a receipt of a NACK both nodes of the faulty link are suspected as bad nodes. As mentioned earlier, this effect is attenuated by the presence of a larger number of diverse flows which contain bad nodes with a variety of good neighbors. These results suggest the effectiveness of Hermes in larger network scenarios.

Figure 7 (b) shows the recommender trustworthiness values, $T^R_{i,j}$, that node i places on nodes j. One sees that nodes 11-30 are correctly identified as bad recommenders by all other nodes that were able to form acceptable recommender trustworthiness values for them. The remaining nodes are correctly identified as good recommenders by the majority of the nodes. We note that there are some false positives in the recommender trustworthiness values. However, the existence of false positives T^R is acceptable as long as the correct opinions P are formed, which is the case here.

9. CONCLUSION

We presented a robust cooperative trust establishment scheme for MANETs, which is designed to improve the reliability of packet forwarding over multi-hop routes, particularly in the presence of malicious nodes. The proposed scheme extends the Hermes framework introduced in [2] in several important ways. In the extended Hermes scheme, first-hand information for non-neighbor nodes is obtained via feedback from acknowledgements sent in response to data packets. The extended Hermes exploits information sharing among nodes to accelerate the convergence of trust establishment procedures. Second-hand trust information is obtained via recommendations from cooperative nodes. The trustworthiness of the recommendations and recommenders is evaluated. The concept of trustworthiness is then extended to the notion of an *opinion* that a given node has about the forwarding behavior of any arbitrary node by combining first-hand and second-hand trust information.

The proposed extensions to Hermes allow nodes to form accurate opinions for any network node and provides robustness against the propagation of false trust information by malicious nodes. 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 either when they operate separately or form collusions. We presented simulation results which demonstrate the effectiveness of the extended Hermes scheme in distinguishing among malicious and non-malicious nodes in a variety of network scenarios involving nodes that are malicious both with respect to packet forwarding and trust propagation.

Acknowledgment

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

10. REFERENCES

[1] L. Eschenauer, V. D. Gligor, and J. Baras, “On trust establishment in mobile ad-hoc networks,” in *Proc. Security Protocols Workshop*, vol. 2845, pp. 47–66, LNCS, April 2002.

[2] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, “A Quantitative Trust Establishment Framework for Reliable Data Packet Delivery in MANETs,” in *In Proceedings of the 3rd ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'05)*, pp. 1–10, November 2005.

[3] L. Zhou and Z. J. Haas, “Securing Ad Hoc Networks,” *IEEE Networks Special Issue on Network Security*, November 1999.

[4] P. Papadimitratos and Z. J. Haas, “Secure message transmission in mobile ad hoc networks,” *Elsevier Ad Hoc Networks Journal*, vol. 1, Jan/Feb/March 2003.

[5] T. Jiang and J. S. Baras, “Ant-based Adaptive Trust Evidence Distribution in MANET,” in *Proceedings of the 2nd International Workshop on Mobile Distributed Computing (MDC)*, March 2004.

[6] S. Buchegger and J.-Y. L. Boudec, “A Robust Reputation System for P2P and Mobile Ad-hoc Networks,” in *Proceedings of the 2nd Workshop on Economics of Peer-to-Peer Systems*, June 2004.

[7] A. A. Pirzada and C. McDonald, “Establishing trust in pure ad-hoc networks,” in *Proceedings of the 27th Australasian Computer Science Conference (ACSC04)*, pp. 47–54, January 2004.

[8] G. Theodorakopoulos and J. S. Baras, “Trust Evaluation in Ad-hoc Networks,” in *Proceedings of the 2004 ACM workshop on Wireless Security (WiSe '04)*, pp. 1–10, 2004.

[9] L. Buttyan and J.-P. Hubaux, “Stimulating Cooperation in Self-Organizing Mobile Ad Hoc Networks,” *Mobile Networks and Applications*, vol. 8, no. 5, pp. 579–592, 2003.

[10] L. Capra, “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.

[11] T. Jiang and J. S. Baras, “Autonomous Trust Establishment,” in *Proceedings of the 2nd International Network Optimization Conference*, 2005.

[12] J. Baras and T. Jiang, “Cooperative Games, Phase Transition on Graphs and Distributed Trust in

MANET," in *Proceedings of the 43rd IEEE Conference on Decision and Control*, June 2004.

[13] S. Marsh, *Formalizing Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.

[14] Y. C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceedings of the ACM MobiCom '02*, ACM SIGMOBILE, September 2002.

[15] P. Papadimitratos and Z. J. Haas, "Secure routing for mobile ad hoc networks," in *SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS) 2002*, January 2002.

[16] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, "Highly secure and efficient routing," in *Proc. IEEE Infocom 2004*, March 2004.

[17] I. Avramopoulos, H. Kobayashi, R. Wang, and A. Krishnamurthy, "Amendment to: Highly secure and efficient routing." unpublished technical note, February 2004.