Ad Hoc Networks xxx (2008) xxx–xxx

Contents lists available at ScienceDirect

Ad Hoc Networks

journal homepage: www.elsevier.com/locate/adhoc

E-Hermes: A robust cooperative trust establishment scheme for mobile ad hoc networks

Charikleia Zouridaki [a], Brian L. Mark [a,*], Marek Hejmo [a], Roshan K. Thomas [b]

[a] Department of Electrical and Computer Engineering, George Mason University, Fairfax, VA 22030, USA
[b] SPARTA, Inc., 5875 Trinity Parkway, Suite 300, Centreville, VA 20120, USA

ARTICLE INFO

Article history:
Received 29 December 2006
Received in revised form 6 October 2008
Accepted 8 October 2008
Available online xxxx

Keywords:
Security
Trust establishment
Reliability
Performance
Routing

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 challenging. 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 with respect to both 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, called *E-Hermes* (Extended-Hermes), which 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. At the same time, the scheme is robust in the presence of 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 second-hand trust information obtained from the propagation of recommendations. This approach 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

* Corresponding author. Tel.: +1 703 993 4069; fax: +1 703 993 1601.
E-mail addresses: charikleia@gmail.com (C. Zouridaki), bmark@gmu.edu (B.L. Mark), mhejmo@gmail.com (M. Hejmo), roshan.thomas@sparta.com (R.K. Thomas).

1570-8705/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.adhoc.2008.10.003

neighbor nodes to non-neighbor nodes by employing a secure acknowledgement protocol.

The main contribution of the present paper[1] is a trust establishment scheme for MANETs, which addresses the propagation of false trust information with respect to packet forwarding behavior. The proposed E-Hermes scheme 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 an acknowledgement scheme for first-hand trust information with respect to non-neighbor nodes and a recommendation scheme that is robust against the propagation of false trust information by malicious nodes. The proposed scheme, in conjunction with a routing protocol based on the computed trust metrics should lead to improved packet delivery in the presence of misbehaving nodes.

The remainder of the paper is organized as follows: Section 2 reviews related work on trust establishment in ad hoc networks and sets the context for the present paper. Sections 3 and 4 discuss the core concepts and advances of the paper. Section 5 addresses the security properties of the proposed trust establishment scheme. Section 6 presents results from simulation experiments that demonstrate the robustness and key properties of the proposed scheme. Finally, the paper is concluded in Section 7.

2. Background and scope of work

2.1. 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 [4] along the lines suggested in [1]. Others have approached trust establishment based on the use of a Bayesian framework [5,2]. 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 [5]. The Hermes scheme presented in [2] builds on the Bayesian approach by incorporating the notion of statistical confidence associated with a trust value.

In [6], 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 relates to trust management was explored in [7] and a semi-ring approach was suggested to evaluate trust and confidence along network paths. In [8], 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 [9], 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 [10,11], which relies on the introduction of pre-trusted agents and a public key infrastructure.

2.2. Hermes framework

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 [6] 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 qualitative or abstract notions of trust, the Hermes framework provides metrics and mechanisms for establishing trust quantitatively with respect to the objective of reliable packet delivery.

The majority of papers related to MANET security focus on securing the route discovery phase of an ad hoc routing protocol. By contrast, the Hermes framework is intended to provide the means to thwart a class of attacks on packet delivery in MANETs during the data transmission phase rather than the route discovery phase. Most of the well-known MANET routing attacks discussed in the literature, such as the wormhole and Sybil attacks, are attacks on the route discovery phase of a routing protocol. Various authors have proposed schemes for avoiding such attacks [12,13] in the route discovery phase.

The Hermes scheme is needed because even if routes are discovered correctly by means of a secure routing protocol, nodes can misbehave during the data transmission phase even if the route is a valid one. Most of the secure routing protocols in the recent literature do not deal with such attacks that occur during the data transmission phase, i.e., packet dropping and packet misforwarding. Moreover, an insider node may behave correctly during the route discovery phase, but then begin misbehaving during the data transmission phase. Secure routing protocols generally do not provide any defense against such attacks.

2.3. Overview of Hermes trust establishment

The notion of trust and trust relationships have been studied extensively in the literature [14]. 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, whereas second-hand information is obtained via third parties. The literature discusses the conveyance of second-hand information through a variety of schemes such as recommendations [6,15–18]. In

[1] A preliminary version of this work was presented in [3].

Hermes [2], opinions represent a 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 original Hermes scheme. For further details, the reader is referred to [2]. 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 sent for forwarding 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 $0 \leqslant t \leqslant 1$. 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 [5].

The confidence value, c, associated with the trust value t is defined as follows:

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

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. The confidence metric is important because a sufficient number of observations must be collected before the empirical trust value t can be considered statistically meaningful. Due to the time-varying, unreliable, and asymmetric characteristics of wireless links and also node mobility, a node X may observe that its downstream neighbor node Y received a packet sent to it by node X, but fail to observe that node Y subsequently forwarded the packet on to node Z, the downstream neighbor of node Y. Such errors will incorrectly bias the value of the counter A, but can be treated as random noise which will be averaged out when the counter M is sufficiently large, i.e., when c is sufficiently close to 1.

At a given time instant a 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 *trustworthiness* metric characterizes a pair (t, c) of trust and confidence values into a single value to facilitate trust-based decisions. The trustworthiness associated with a pair (t, c) is defined as [2]

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

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

$$T_{\text{def}} \triangleq T(0.5, 0), \tag{4}$$

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 with respect to the confidence level. The concept of acceptability is used in calculating second-hand trust information (see Section 4.1). The pair (t, c) is *acceptable* if a sufficient amount of observation data has been accumulated such that $c > c_{\text{acc}}$. We remark that each node may choose a different value of c_{acc} to implement its own policy in determining the acceptability of trustworthiness values. The choice of c_{acc} is a tradeoff between accuracy and convergence time. If c_{acc} is large (i.e., close to one), the trustworthiness values obtained will be more accurate, but the convergence time will be longer.

2.4. Scope of E-Hermes

The E-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. Even when the routing protocol is not secure, the E-Hermes scheme can mitigate the effects of routing attacks such as the wormhole and Sybil attacks, as discussed in Section 5. 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 the Hermes scheme 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 a novel mechanism for deriving trustworthiness values for non-neighbor nodes based on first-hand information from acknowledgements, as opposed to relying on second-hand recommendations alone. These extensions to Hermes were introduced in an earlier paper [3]. The present paper goes beyond [3] by including the following: (1) a more complete formulation of the E-Hermes scheme; (2) a more detailed analysis of security properties; (3) a discussion of the behavior of E-Hermes under various attack scenarios; (4) a discussion of the communication and computational overhead; (5) a simpler formulation of the acknowledgement scheme; and (6) additional simulation results. The key security properties provided by the E-Hermes scheme, beyond what is provided in the original Hermes scheme [2], are summarized as follows:

(i) Ability to capture independent packet forwarding and trust propagation misbehaviors.
(ii) Resilience to the presence of bad nodes and bad recommenders.
(iii) Resilience to attacker placement.

In addition, the E-Hermes scheme provides faster convergence and more robustness than the original Hermes scheme due to the gathering of first-hand trust information for non-neighbor nodes via the proposed acknowledgement scheme.

Please cite this article in press as: C. Zouridaki et al., E-Hermes: A robust cooperative trust establishment scheme ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.10.003

2.5. Attacker model

In this paper, we assume an attacker model in which a node may drop, misroute or replay data packets that it is supposed to forward under a given routing protocol. A node that performs this type of attack with a certain statistical regularity is referred to as a *bad node*. A node that forwards the majority of its packets correctly, with statistical regularity, is referred to as a *good node*. Analogously, we define a *bad recommender* as a node that incorrectly propagates recommendations with a certain statistical regularity. Conversely, a node that propagates recommendations correctly, with high statistical regularity, is a *good recommender*.

The above notations can be made more precise by modeling the frequency with which a node causes a fault in terms of probabilities. More specifically, for each node $i \in \mathcal{N}$, let B_f^i denote the probability the node i incorrectly forwards a data packet and let B_t^i denote the probability that it incorrectly propagates a recommendation.

Definition 1. Node i is defined to be *bad* if $B_f^i < T_{\text{def}}$. Conversely, node i is *good* if $B_f^i > T_{\text{def}}$.

Definition 2. Node i is defined to be a *bad recommender* if $B_t^i < T_{\text{def}}$. Conversely, node i is a *good recommender* if $B_t^i > T_{\text{def}}$.

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 discussion in Section 5, but does not represent any limitation in the E-Hermes scheme itself. In the E-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.

3. First-hand trust evaluation

In this section, we present a new scheme for gathering first-hand trust information from non-neighbor nodes using acknowledgements.

3.1. Wireless channel snooping nodes

In the Hermes scheme, nodes evaluate the trustworthiness of their neighbors by snooping the wireless channel. It is assumed that the nodes are equipped with omnidirectional antennas and that they do not employ dynamic power control. We use the term *fault* to denote an event in which a node fails to forward a packet correctly to its next hop. A fault may occur due to malicious or non-malicious misbehavior of a node. Non-malicious packet forwarding misbehavior may be due to such phenomena as network congestion, node mobility, or node malfunction.

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.

Definition 3. The expiry of the timer indicates that an incorrect packet forwarding event has occurred. We refer to this event as a *fault*. When a fault attributed to node y occurs, the counter M_y is incremented by one and A_y is not updated. In this case, we say that node x *penalizes* node y.

3.2. Acknowledgement scheme

We now propose an acknowledgement scheme to evaluate first-hand trust when the underlying routing protocol is based on source routing, such as DSR [19]. To incorporate trust into source routing, nodes must establish trust for non-neighbor nodes. We remark that for distance vector routing protocols, such as AODV [20], it is sufficient to establish trust only for neighbor nodes.

To obtain first-hand information from non-neighbor nodes, we propose an acknowledgement 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:

$$M_{y_i} \leftarrow M_{y_i} + 1, \quad 1 \leqslant i \leqslant n-1. \tag{5}$$

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

If node x receives an acknowledgement (ACK) packet from node y_1 within the timeout interval, it forwards the ACK to its upstream neighbor and updates the A-counters for all of the downstream intermediate nodes as follows:

$$A_{y_i} \leftarrow A_{y_i} + 1, \quad 2 \leqslant i \leqslant n-1, \tag{6}$$

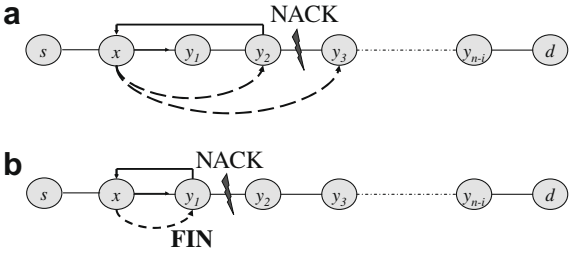


Fig. 1. Processing of NACKs.

Please cite this article in press as: C. Zouridaki et al., E-Hermes: A robust cooperative trust establishment scheme ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.10.003

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 3.1.

In the case when an ACK is not received within the timeout interval, the node creates a negative acknowledgement (NACK) packet and sends the NACK to its upstream neighbor on the path. This situation is illustrated in Fig. 1, where the dashed arrow from node x to node y indicates that x penalizes y for the fault and the solid arrow indicates the transmission of a NACK. Fig. 1a illustrates the subcase where node x receives a NACK from node y_2 and Fig. 1b illustrates the subcase when node x receives a NACK from node y_1. The distinction between the two subcases is that y_1 is a first intermediate node (FIN), i.e., the first node downstream from the recipient of the NACK.

Consider the subcase in which the NACK originates from node $y_i, 2 \leqslant i \leqslant n-1$. Here, 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. 1a). Therefore, node x penalizes both nodes of the link. To avoid penalizing the nodes downstream from y_{i+1}, the M-counters for these nodes are decremented by one

$$M_{y_j} \leftarrow M_{y_j} - 1, \quad i+2 \leqslant j \leqslant n-1. \tag{7}$$

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

$$A_{y_j} \leftarrow A_{y_j} + 1, \quad 1 \leqslant j \leqslant i-1. \tag{8}$$

In the subcase where the NACK originates 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, since y_1 is the FIN, x can monitor the forwarding behavior of y_1 at the MAC layer, allowing it to isolate the fault to y_2. To avoid penalizing the nodes downstream from y_2, the M-counters for the nodes $y_3, \ldots, y_{n-1}$ are decremented by one:

$$M_{y_i} \leftarrow M_{y_i} - 1, \quad 3 \leqslant i \leqslant n-1. \tag{9}$$

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

$$M_{y_i} \leftarrow M_{y_i} - 1, \quad 2 \leqslant i \leqslant n-1. \tag{10}$$

In both subcases, node x forwards the NACK to its upstream neighbor. In subcase (b), node x verifies whether or not node y_1 correctly forwarded packet p based on snooping the wireless channel. However, in some networks, channel snooping may be vulnerable to certain types of attack (cf. Section 5.6). In such networks, subcase (b) should be handled similarly to subcase (a), i.e., *both* nodes y_1 and y_2 are be penalized upon receipt of a NACK originating from y_1. Note that the A and M counters are maintained only for downstream nodes. Upon receipt of a NACK, both ends of the associated link are penalized (except in subcase (b), with channel snooping) even if only one of the nodes may be responsible for the fault. However, this effect diminishes as more observation data involving the two nodes with respect to different flows is accumulated over time.

3.3. 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))

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

from which the trustworthiness value T_y can be computed via (3).

3.4. Authentication of packets

Authentication of every data, recommendation, ACK, and NACK packet is required to protect the network against modification and impersonation attacks. We adopt a variation of the scheme proposed in [21] for hop-by-hop authentication, based on hash chains. We assume that the nodes have already established a set of pairwise keys using a key management protocol [22,23]. If a secure routing protocol is in place, the keys established for secure routing can be used to secure the E-Hermes scheme. Let $K_{i,j}$ denote the shared key between node i and node j. Consider a path $R = \{s, a_1, a_2, \ldots, a_{n-1}, a_n = d\}$, where $n \geqslant 2$, from source node s to destination node d. Let k denote the sequence number of a given data packet that is forwarded along the path R.

3.4.1. Data and recommendation packets

As in [21], the authentication field, $\mathscr{A}$, of a data packet with data field $\mathscr{D}$ sent along route R, consists of a sequence of message authentication codes (MACs):

$$\mathscr{A} = [\mathscr{M}_n, \mathscr{M}_{n-1}, \ldots, \mathscr{M}_1].$$

The MACs are defined as follows:

$$\mathscr{M}_n = f(K_{s,a_n}, \mathscr{D}),$$

and for $i = 1, \ldots, n-1$:

$$\mathscr{M}_i = f(K_{s,a_i}, [\mathscr{D}, \mathscr{M}_n, \ldots, \mathscr{M}_{i+1}]),$$

where $f(K, \mathscr{X})$ denotes the function that produces a MAC from the key K and data $\mathscr{X}$. The authentication field allows each intermediate node to authenticate the packet and protects against malicious intermediate nodes that try to tamper with the MAC field of a downstream node. In the E-Hermes scheme, the intermediate nodes along the route need to be able to authenticate data packets in order to collect packet statistics to derive first-hand trust information.

Recommendation request and reply packets are not used to collect first-hand trust information. Therefore, for recommendation packets, it suffices for the authentication field to consist only of a single MAC computed using the

Please cite this article in press as: C. Zouridaki et al., E-Hermes: A robust cooperative trust establishment scheme ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.10.003

shared key between the recommender and the source of the recommendation request.

3.4.2. Control packets

In [21], the authentication fields of each ACK or NACK control packet are designed to satisfy three properties: (i) forging is impractical, (ii) an ACK or NACK verified at one non-faulty node on a path, also verifies at all non-faulty nodes on the path, (iii) authentication of node identities. We extend the scheme of [21] to provide a fourth property: (iv) authentication of whether the packet is an ACK or a NACK.

A one-way hash function $h(\cdot)$ and hash chains of length three, associated with each control packet, are used to guarantee these properties. For packet k and intermediate node a_i, a hash chain is used to authenticate ACK packets traveling upstream on route R. Let $\alpha_i^0(k)$ denote the initial element of the "ACK" hash chain for node a_i for $i = 1, \ldots, n$. The hash chain element $\alpha_i^0(k)$ is constructed by concatenating the key $K_s^{a_i}$, the sequence number k, and the element 0. The second and third elements in the ACK hash chain associated with packet k and node a_i are

$$\alpha_i^1(k) \triangleq h[\alpha_i^0(k)] \quad \text{and} \quad \alpha_i^2(k) \triangleq h[\alpha_i^1(k)],$$

respectively.

We extend the scheme of [21] by defining a three-element hash chain associated with packet k and node a_i is defined for the authentication of NACK packets for $i = 1, \ldots, n-1$. Note that a "NACK" hash chain element is not required for the destination node a_n, since it never transmits NACK packets. Let $\eta_i^0(k)$ denote the initial element of the NACK hash chain for node a_i. The hash chain element $\eta_i^0(k)$ is constructed by concatenating the key $K_s^{a_i}$, the sequence number k, and the element 1. The second and third elements in the ACK hash chain associated with packet k and node a_i are

$$\eta_i^1(k) \triangleq h[\eta_i^0(k)] \quad \text{and} \quad \eta_i^2(k) \triangleq h[\eta_i^1(k)],$$

respectively. When node s transmits data packet k along route R, it concatenates the third elements of the ACK and NACK hash chains associated with the intermediate nodes, i.e.,

$$\alpha_1^2(k), \alpha_2^2(k), \ldots, \alpha_n^2(k),$$
$$\eta_1^2(k), \eta_2^2(k), \ldots, \eta_{n-1}^2(k).$$

As packet k is forwarded along the path R, each intermediate node a_i $(1 \leqslant i \leqslant n-1)$ extracts and stores the hash chain elements corresponding to the downstream nodes.

The scheme of [21] is vulnerable to a certain attack because only a single hash chain, for both ACKs and NACKs is used. Consider the path $R = \{s, a_1, a_2, a_3, a_4 = d\}$. Since only a single hash-chain is used to represent both ACKs and NACKs, node a_1 can create a NACK and forward it to the source such that it will believe that node a_2 constructed a NACK for link (a_2, a_3). Consequently, nodes a_2 and a_3 will be penalized erroneously. With our proposed extension, node a_1 cannot launch the aforementioned attack because it cannot create a valid NACK packet attributed to node a_2.

4. 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 confidence value $c_{i,m}$ is below c_{acc}. 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*.

4.1. Processing recommendations

Definition 4. 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 called the *recommender*.

Node i seeks recommendations on a node m when the confidence it has computed for m is below c_{acc} (see Section 2.3). Node i discriminates among multiple recommenders by evaluating a metric called *recommender trustworthiness*.

Definition 5. *Recommender trustworthiness* $T_{i,j}^R$ is the trustworthiness that node i places on recommender node j as a measure of how reliably node j propagates trustworthiness information.

Definition 6. A node j is *considered a good recommender* by node i when the recommender trustworthiness $T_{i,j}^R$ that i places on recommender j exceeds T_{def}.

Definition 7. A node j is *considered a bad recommender* by node i when the recommender trustworthiness $T_{i,j}^R$ that i places on recommender j is smaller than T_{def}.

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 confidence value smaller than c_{acc}. 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 confidence value $c^R < c_{\text{acc}}$, 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 $f \leqslant d$ recommendations due to network conditions or lack of willingness to respond to the request. Additionally, when node j has a confidence value for node m smaller than c_{acc}, 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.

After receiving a set $R_m = \{T_{j,m} : j \in D\}$ of recommendations for node m, node i performs the following steps. If the confidence value $c_{i,m}$ is smaller than c_{acc}, node i calculates a "temporary" trustworthiness value $\tilde{T}_{i,m}$, which is taken as the maximum trustworthiness value $T_{j,m}$ among the recommenders $j \in D$, i.e.,

$$\tilde{T}_{i,m} = \max\{T_{j,m} : j \in D\}. \quad (11)$$

The value $\tilde{T}_{i,m}$ is used for routing or any other network-related decisions until subsequent updates result in the value of $c_{i,m}$ exceeding c_{acc}.

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

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

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^R and M^R, where A^R counts the number of times for which the RC-test succeeds and M^R counts the total number of times that the RC-test is applied. The A^R and M^R counters are then used to calculate the *recommender trustworthiness* $T^R_{i,j}$ 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 3.2, when node m sends a NACK upstream, the source node i attributes the fault both to m and its downstream neighbor. 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 for 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.

4.2. Calculation of opinion

We 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 $P_{i,m}$. The definition for the opinion that any node i has for another node m is given as follows:

$$P_{i,m} \triangleq \max_{j \in \Gamma}\{\omega_{i,j} T_{j,m}\}, \ \text{for } P_{j,m} \neq T_{\text{def}}, \quad (12)$$

where

$$\omega_{i,j} = \begin{cases} T^R_{i,j}, & i \neq j, \\ 1, & i = j. \end{cases} \quad (13)$$

and Γ is the set of recommenders in D that have passed the RC-test.

Nodes are judged to be *good* or *bad* on the basis of the opinion value.

Definition 8. A node j is *considered good* by node i when the opinion $P_{i,j} > T_{\text{def}}$.

Definition 9. A node j is *considered bad* by node i when the opinion $P_{i,j} < T_{\text{def}}$.

5. Security evaluation

We analyze the resistance of E-Hermes to (1) incorrect data packet forwarding, and (2) incorrect propagation of trust information attacks. As discussed in Section 1, during the data transmission phase authorized or insider nodes may consistently drop, misroute, or replay data packets. The Hermes scheme identifies 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. Note that we do not distinguish among the various types of data packet forwarding misbehaviors, i.e., packet dropping, misrouting, and replay attacks.

We consider the response of the E-Hermes schemes in various attack scenarios, with respect to a single flow. As we shall see, in each case, the E-Hermes scheme successfully penalizes the bad nodes and bad recommenders. The upstream neighbor of the bad node will also be penalized even if it happens to be a good node, since the ACK scheme penalizes bad *links* along the route. In general, however, the upstream neighbor will be credited as a good node with respect to other flows. Routing diversity ensures that a good node will be recognized as bad only with low probability.

5.1. Bad nodes

Fig. 2 illustrates the response of Hermes to packet forwarding misbehavior from a single bad node, labelled X, on a route $R_1 = \{Y_2, Y_1, o, X, Z_2, \ldots\}$ corresponding to flow f_1. The case of multiple bad nodes along a given path is similar. In Fig. 2, node X incorrectly forwards data packets on flow f_1 with probability B^X_f, where $0 < B^X_f \leqslant 1$. Since node o is a neighbor of X, it obtains first-hand information about the packet-forwarding behavior of node X at the MAC layer. The nodes upstream of node o, i.e., nodes Y_1-Y_3, infer first-hand trust information from the NACKs initiated by node o. Since node o is a neighbor of node Y_1, node Y_1 is able to verify the correct forwarding behavior of node o. Thus, upon receiving a NACK from node o, node Y_1 penalizes node X. On the other hand, upon receiving a NACK initiated by node o, nodes Y_1 and Y_2 penalize both nodes o and X. In this case, node o can be recognized as a good node only through other flows in which node o is not penalized.

In our attacker model, we have assumed that when a node forwards packets correctly, it also propagates ACK and NACK packets correctly, whereas when a node incorrectly forwards data packets, it does not initiate NACK packets. Consider a scenario in which an attacker node X drops or misroutes data packets, and initiates NACK

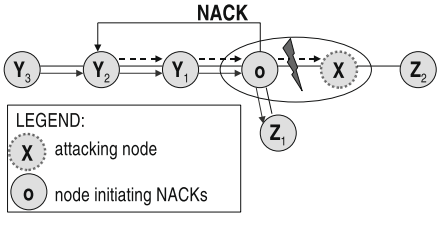


Fig. 2. Single attacker node.

packets in an attempt to accuse its downstream neighbor of incorrect data packet forwarding. Such an attack does not benefit the attacker, since the E-Hermes scheme penalizes both ends of the link at fault. As a result, all non-neighbor upstream nodes of attacker X on the route will penalize node X.

5.2. Bad recommenders

The E-Hermes scheme makes use of trustworthiness information exchanged among nodes through recommendations. An obvious attack on the E-Hermes scheme would be for a given node to propagate false trustworthiness information, i.e., the node propagates a trustworthiness value that is different from the value that it would compute if it were following the Hermes scheme. The RC-test (see Section 4.1) ensures that recommendations are accepted only when the recommended trustworthiness value is sufficiently close to the *first-hand trustworthiness value* computed by the node that asked for the recommendations, provided that $c > c_{\text{acc}}$ for this node. If $c \leqslant c_{\text{acc}}$, the node only temporarily accepts the maximum value from among all the recommenders. Because of this, bad recommender nodes can be identified by the scheme.

Next, consider the case when a node is falsely categorized as a bad recommender. Due to the RC-test, this categorization does not affect the trust establishment process for a node, since "bad" recommendations are discarded. Fig. 3 illustrates an example of this type of scenario. Source node Y_2 establishes route $R_1 = \{Y_2, Y_1, o, X, Z_2, \ldots\}$ for its flow f_1. Node X forwards data packets incorrectly with probability $0 < B_f^X \leqslant 1$. X's upstream neighbor node o will initialize NACKs for all packets that are not acknowledged by node X. Node Y_2 will penalize both nodes o and X. Now suppose that node Y_3 establishes route $R_2 = \{Y_3, Y_1, o, Z_1, \ldots\}$ for its flow f_2. Node Y_3 sees node o as a good node. If nodes Y_2 and Y_3 exchange recommendations about node o at this point in time, they will consider each other as bad recommenders. However, this will not affect their trust establishment processes. Moreover, if Y_2 collects more observations of node o from flows that do not traverse node X, eventually node Y_2 will compute a high trustworthiness value for node o.

5.3. Collusion of bad node and recommender

Fig. 4 illustrates a colluding attack involving a bad node and a bad recommender (yet good node) on a route. Source node Y_2 establishes route $R = \{Y_2, Y_1, w, X, Z_1, \ldots\}$ for its flow f. Node X forwards data packets incorrectly with probability B_f^X, while node w propagates trustworthiness values for node X higher than the value that a Hermes-compliant node would compute. In doing this, node w is attempting to persuade the upstream nodes on the route that node X is a good node. Node w does not initiate NACKs for the unacknowledged packets that node X incorrectly forwards; otherwise, it would be contradicting itself. Instead, node X would send NACKs itself, while dropping or misrouting packets. In this case, all non-neighbor upstream nodes of the attacker X on the route would correctly penalize the attacker node X.

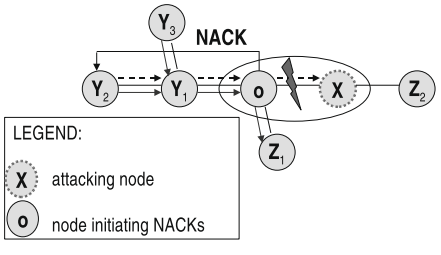


Fig. 3. Bad recommender false positive.

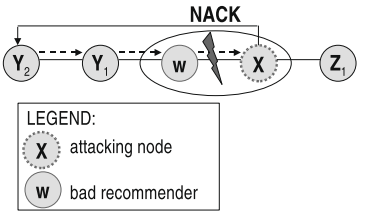


Fig. 4. Attacker node and colluding bad recommender.

5.4. Wormhole attack

If a wormhole attack occurs and no mechanism is in place to prevent this attack during the route discovery phase (cf. [12]), the E-Hermes scheme will still be able to identify the misbehaving nodes involved in the wormhole. For example, suppose that two colluding nodes X and Y form a wormhole. They may be connected by a wired link or a wireless link formed using a directional antenna. Thus, nodes X and Y could form part of a route even though they are not neighbors. Now suppose packets are sent on a route which includes node X and Y. If the packets are not forwarded correctly by nodes X and Y through the wormhole, the ACK scheme of E-Hermes will penalize both nodes X and Y.

5.5. Sybil attack

In the Sybil attack, a node impersonates one or more of the other nodes in the network. This is an attack on the authentication scheme and can really only be addressed using cryptographic techniques (cf. [13]). Suppose a given node X launches a Sybil attack by impersonating another node Z on given route. Under the E-Hermes scheme, if node X drops packets sent along this route, then node Z will be penalized. On the other hand, node X is also penalized with respect to this route, since it claims to be node Z.

5.6. Attackers with directional antennas

Thus far, we have implicitly assumed that the nodes are equipped with omnidirectional antennas and that they do not employ dynamic power control. However, an outside adversary could use a directional antenna to launch an attack. E-Hermes can deal with this type of attack as follows. Suppose nodes X and Y are neighbors along a route and

Please cite this article in press as: C. Zouridaki et al., E-Hermes: A robust cooperative trust establishment scheme ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.10.003

node Y is an attacker equipped with a directional antenna. Now suppose node Y forwards packets in the direction of node X such that node X believes that node Y forwarded these packets correctly, when in actual fact, node Y does not forward these packets to the next node on the path.

In this case, the first-hand trust evaluation based on neighbor observations (i.e., channel snooping) in E-Hermes will be foiled. However, first-hand trust evaluation for non-neighbor nodes in E-Hermes, which is based on acknowledgments, will identify node Y correctly as the culprit. To see this, note that Y can attempt to deceive node X in one of three possible ways: (i) send a forged ACK to X; (ii) create a NACK and then send it to X; or (iii) send neither an ACK nor a NACK to X. The hash-chain scheme discussed in Section 3.4.2 precludes action (i) from being successful. In case (ii), both node Y and its downstream neighbor will be penalized. In case (iii), the acknowledgement timer of node X will expire, causing node X to penalize node Y and its downstream neighbor. Thus, node Y cannot avoid being penalized for the attack. Hence, to deal with directional antenna attacks, the E-Hermes scheme should place more weight on the second form of first-hand trust or avoid making use of neighbor observations altogether.

6. Performance evaluation

In this section, we present some representative performance results of E-Hermes obtained from simulation experiments.

6.1. Simulation methodology

The simulation experiments were implemented using MATLAB and are intended to evaluate the performance of E-Hermes under various network and attack scenarios. In the simulation 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.

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. In particular, traffic flows are generated as a function of the number of network nodes and the minimum and maximum number of nodes allowed on a route with no routing loops. 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.

The bad nodes may be neighbors or non-neighbors. The number of traffic flows generated in the simulation scenarios presented in this section is relatively small. 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 and diverse set of traffic flows, all nodes should be able to form valid opinions for every other node in the network.

6.2. Static node behavior

We consider a simulation scenario consisting of 10 nodes and 8 random traffic flows are established along different paths. 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. 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 E-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 in (3) is set as $r = \sqrt{2/9}$. Finally, the RC-test threshold η is set to 0.1.

Fig. 5 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. 5 (a) illustrates the opinion values, $P_{i,j}$, obtained by the scheme without the use of recommendations. Nodes that interacted with nodes 2 and 7 correctly identified them as bad nodes. Nodes that interacted with the remaining nodes identified them as good, with two exceptions. The two false positives are attributed to the fact that upon receipt of a NACK both ends of the faulty link are penalized. This effect would be attenuated by the establishment of a more diverse set of flows.

Fig. 5b illustrates the opinion values, $P_{i,j}$ when recommendations are used. Nodes 2 and 7 are correctly identified as bad nodes by all other nodes, except node 6, which is ignorant of their behaviors ($T_{6,2} = T_{6,7} = T_{\text{def}}$). Node 7 has not identified node 2 as a bad node for the same reason. The good nodes are also correctly identified. Comparing (a) and (b) we see that when recommendations are used, nodes form the correct network view much more quickly. We have evaluated the E-Hermes scheme under various attack scenarios by varying the number of bad recommenders and bad nodes, and found that the scheme computes accurate opinions in all cases.

Fig. 5c 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 compute 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 T^R, because only eight flows

Please cite this article in press as: C. Zouridaki et al., E-Hermes: A robust cooperative trust establishment scheme ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.10.003

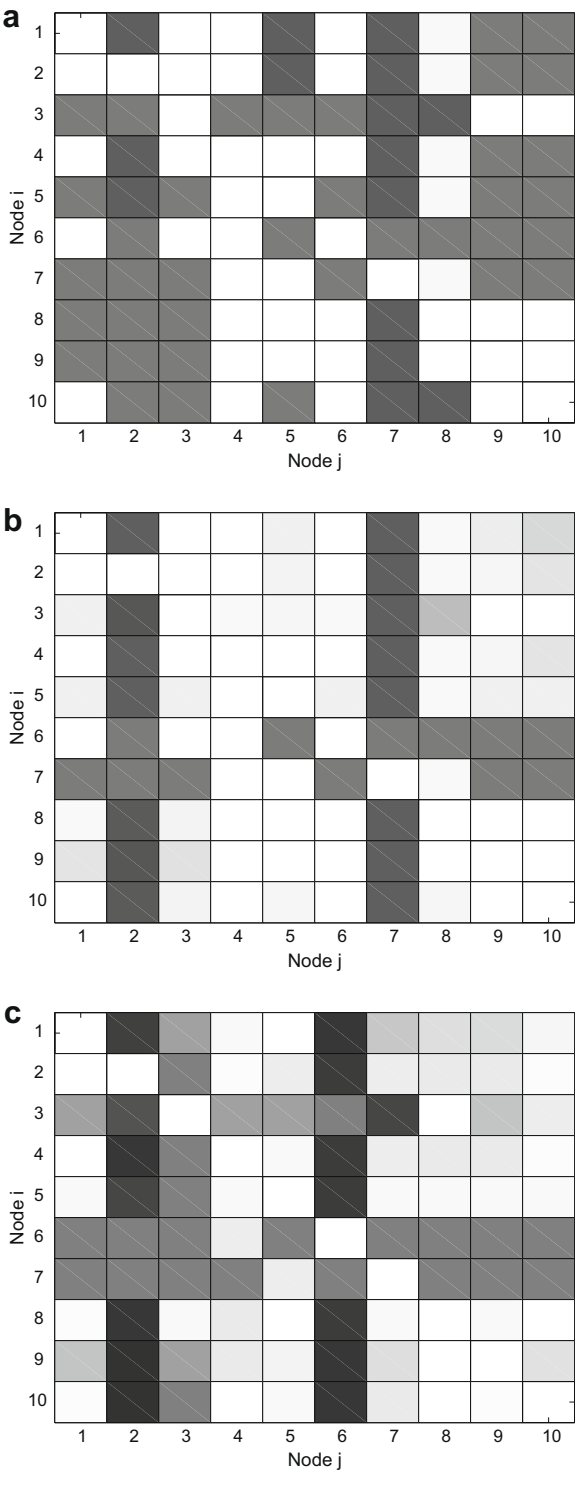


Fig. 5. Network view (a) opinion $P_{i,j}$, without recommendations, (b) opinion $P_{i,j}$, with recommendations, (c) Recommender trustworthiness $T_{Ri,j}$.

are active. As the diversity of flows in the network increases, the accuracy of the opinions computed improves. However, note that the existence of false positives T^R is acceptable, as long as the correct opinions P are formed, which is the case here.

6.3. Dynamic node behavior

Next, we consider a simulation scenario in which the behavior of a node changes dynamically. Eight flows are generated and the source nodes send 100 data packets during each round. The simulation runs for 50 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 Types III–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. 6. E-Hermes 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 $P_{10,7} = 0.86$.

6.3.1. Misbehavior recognition

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

Definition 10. The *misbehavior* (ϵ) *recognition* percentage or *MB(ϵ)-recognition* is the percentage of the nodes in the network that have identified all the misbehaving nodes in the network by computing the opinions $P_{i,m}$ that are within a precision of ϵ from the true node behavior characterized by B_f^i, i.e., $|P_{i,m} - (1 - B_f^i)| < \epsilon$.

We present some performance results of E-Hermes with respect to MB(0.1)-recognition when the percentage of bad nodes and B_f are varied. In particular, the percentage of misbehaving network nodes ranges from 4% to 95%, while B_f ranges from 20% to 100%. The number of bad recommenders is set to 25% of the network nodes. Our simulation runs are intended to evaluate the MB(0.1)-recognition metric and the convergence rate of E-Hermes when the

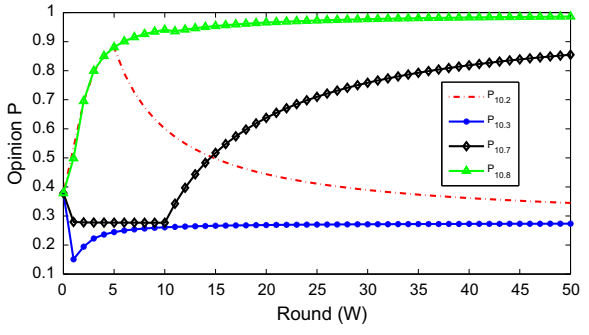


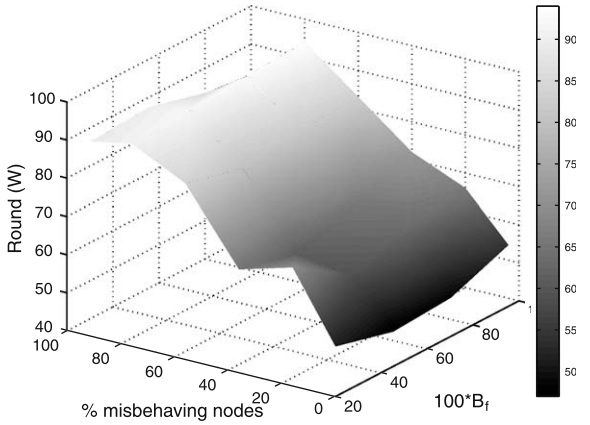
Fig. 6. Opinion that node 10 computes for nodes 2, 3, 7, 8 from round 1 to 50. Nodes 2, 7 change their forwarding behaviors in rounds 5 and 10, respectively.

percentage of bad recommenders and B_f are varied. In particular, the percentage of bad recommenders ranges from 4% to 100%, while B_f ranges from 20% to 100%. The proportion of bad nodes is set to 25% of the network nodes.

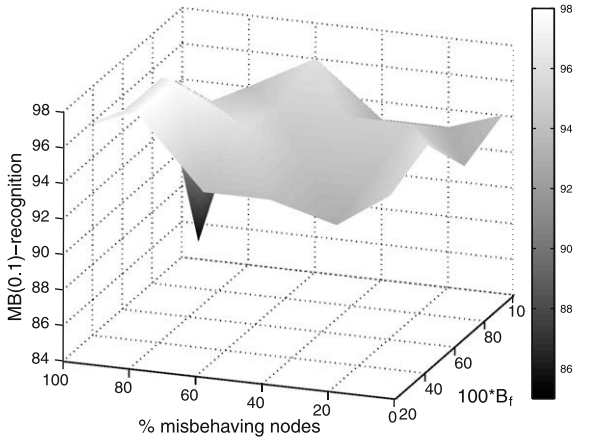
The bad nodes and the bad recommenders are chosen randomly from the set of the network nodes. Thus, a node may exhibit any of the four types of behavior introduced in Section 6.1. The simulated network consists of 30 nodes. Initially one flow is generated and then one flow is added per round. The flows are randomly generated. The number of nodes on a route is set to 7. The non-misbehaving nodes forward all the packets that they receive for forwarding. The good recommenders propagate correct opinions P. The bad recommenders propagate fixed opinion $P = 0.5$ when $B_f \neq 0.5$ and they propagate fixed opinion $P = 0.2$ when $B_f = 0.5$. The source nodes send 100 data packets during each round. The other simulation parameters are set as before. The results are obtained from executing 10 simulation trials for each network scenario.

Fig. 7a illustrates the number of rounds required for E-Hermes to reach a steady state. As expected, the convergence rate depends on the percentage of misbehaving network nodes; the more misbehaving nodes in the network, the longer it takes for E-Hermes to reach a steady state. The value of B_f slightly influences the convergence rate of the proposed trust establishment framework. For example, when 4% of the nodes are misbehaving, E-Hermes requires 50, 47, 48, and 54 rounds to reach steady state when B_f is set to 0.25, 0.5, 0.75 and 1, respectively. When there are 95% of misbehaving network nodes, E-Hermes requires 89, 90, 86, and 76 rounds to reach steady state when B_f is set to 0.25, 0.5, 0.75, and 1 respectively.

Fig. 7b shows the steady-state MB(0.1)-recognition values for E-Hermes. As expected, the MB(0.1)-recognition of E-Hermes is in the range 93–98%, with one exception: When 95% of the nodes are misbehaving and $B_f = 1$, the MB(0.1)-recognition of E-Hermes is 85%. Nonetheless, it should be noted that steady state is reached only after 76 rounds.

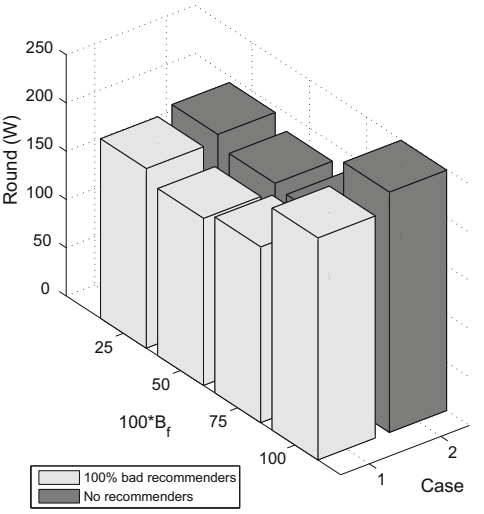


(a) Convergence rate.

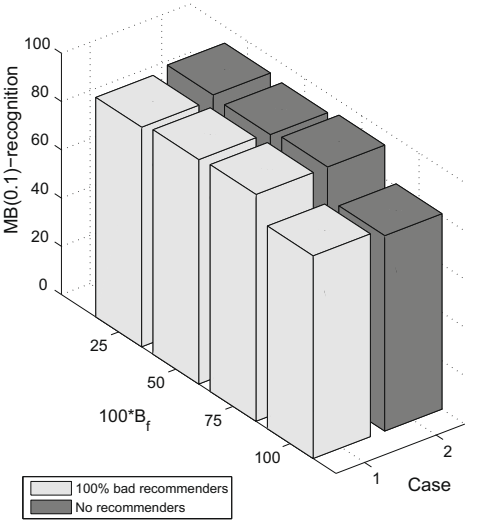


(b) MB(0.1)-recognition.

Fig. 7. E-Hermes performance when the percentage of misbehaving nodes and B_f are varied.



(a) Convergence rate.



(b) MB(0.1)-recognition.

Fig. 8. E-Hermes performance when (1) 100% of nodes are bad recommenders, and (2) recommendations are not exchanged.

Please cite this article in press as: C. Zouridaki et al., E-Hermes: A robust cooperative trust establishment scheme ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.10.003

Fig. 8a compares the number of rounds required for E-Hermes to reach a steady state when (1) 100% of the nodes in the network are bad recommenders, and (2) recommendations are not exchanged in the network. Fig. 8b shows the MB(0.1)-recognition of E-Hermes in steady state when (1) 100% of the nodes are bad recommenders, and (2) recommendations are not exchanged in the network. Thus, the figures show that the exchange of bad recommendations does not undermine the performance of E-Hermes. However, the availability of good recommendations does accelerate the convergence of the trust establishment procedures. If all the nodes in the network are bad recommenders, E-Hermes performs as if no recommenders are present in the network.

7. 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 E-Hermes scheme, first-hand information for non-neighbor nodes is obtained via feedback from acknowledgements sent in response to data packets. The E-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.

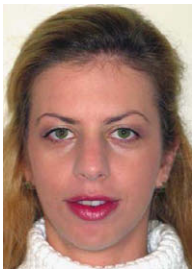
A potential problem arises when a node behaves well with respect to some flows, but behaves badly with respect to other flows. The E-Hermes scheme may not be able to compute accurate trustworthiness values in this case. However, such Byzantine behavior can be addressed by extending the Hermes framework in a different way, as discussed in [24].

Acknowledgement

This work was supported in part by the National Science Foundation under Grants Nos. CCR-0209049 and CCF-0133390.

References

[1] L. Eschenauer, V.D. Gligor, J. Baras, On trust establishment in mobile ad-hoc networks, in: Proceedings of the Security Protocols Workshop, vol. 2845, LNCS, 2002, pp. 47–66.
[2] C. Zouridaki, B.L. Mark, M. Hejmo, R.K. Thomas, Hermes: a quantitative trust establishment framework for reliable data packet delivery in MANETs, Journal of Computer Security 15 (1) (2007) 3–38.
[3] C. Zouridaki, B.L. Mark, M. Hejmo, R.K. Thomas, Robust cooperative trust establishment for MANETs, in: Proceedings of the Third ACM Workshop on Security of Ad Hoc and Sensor Networks (SASN'06), 2006, pp. 23–34.
[4] T. Jiang, J.S. Baras, Ant-based adaptive trust evidence distribution in MANET, in: Proceedings of the Second International Workshop on Mobile Distributed Computing (MDC), 2004.
[5] S. Buchegger, J.-Y.L. Boudec, A robust reputation system for P2P and mobile ad-hoc networks, in: Proceedings of the Second Workshop on Economics of Peer-to-Peer Systems, 2004.
[6] A.A. Pirzada, C. McDonald, Establishing trust in pure ad-hoc networks, in: Proceedings of the 27th Australasian Computer Science Conf. (ACSC'04), 2004, pp. 47–54.
[7] G. Theodorakopoulos, J.S. Baras, Trust evaluation in ad-hoc networks, in: Proceedings of the ACM Workshop on Wireless Security (WiSe'04), 2004, pp. 1–10.
[8] L. Buttyan, J.-P. Hubaux, Stimulating cooperation in self-organizing mobile ad hoc networks, Mobile Networks and Applications 8 (5) (2003) 579–592.
[9] L. Capra, Engineering human trust in mobile system collaborations, in: Proceedings of the 12th ACM SIGSOFT International Symposium on Foundations of Software Engineering, 2004, pp. 107–116.
[10] T. Jiang, J.S. Baras, Autonomous trust establishment, in: Proceedings of the Second International Network Optimization Conference, 2005.
[11] J. Baras, T. Jiang, Cooperative games, phase transition on graphs and distributed trust in MANET, in: Proceedings of the 43rd IEEE Conference on Decision and Control (CDC'04), 2004.
[12] Y.C. Hu, A. Perrig, D.B. Johnson, Wormhole Attacks in Wireless Networks, IEEE Journal of Selected Areas in Communications 24 (2) (2006) 370–380.
[13] D. Glynos, P. Kotzanikolaou, C. Douligeris, Preventing impersonation attacks in MANET with multi-factor authentication, in: Proceedings of the International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WIOPT), 2005, pp. 59–64.
[14] S. Marsh, Formalizing trust as a computational concept, Ph.D. Thesis, University of Stirling, 1994.
[15] A. Abdul-Rahman, S. Hailes, Supporting trust in virtual communities, in: Proceedings of the IEEE Hawaii International Conference on System Sciences, 2000.
[16] B. Yu, M.P. Singh, A social mechanism of reputation management in electronic communities, in: Proceedings of the Fourth International Workshop on Cooperative Information Agents, vol. 1860, LNCS, 2000, pp. 154–165.
[17] P. Resnick, K. Kuwabara, R. Zeckhauser, E. Friedman, Reputation systems, Communications of the ACM 43 (12) (2000) 45–48.
[18] J.P. Hubaux, L. Buttyan, S. Capkun, The quest for security in mobile ad hoc networks, in: Proceedings of the ACM MobiHoc, 2001.
[19] D. Johnson, D. Maltz, Dynamic source routing in ad hoc wireless networks, in: T. Imielinski, H. Korth (Eds.), Mobile Computing, Kluwer Academic Publishers, 1996, pp. 153–181 (Chapter 5).
[20] C. Perkins, E. Belding-Royer, S. Das, Ad-hoc on-demand distance vector (AODV) routing, in: IETF RFC 3561.
[21] I. Avramopoulos, H. Kobayashi, R. Wang, A. Krishnamurthy, Highly secure and efficient routing, in: Proceedings of the IEEE Infocom 2004, 2004.
[22] L. Zhou, Z.J. Haas, Securing Ad Hoc Networks, IEEE Networks Special Issue on Network Security 13 (6) (1999) 24–33.
[23] N. Asokan, P. Ginzboorg, Key agreement in ad-hoc networks, Computer Communications Journal 23 (17) (2000) 1627–1637.
[24] C. Zouridaki, B.L. Mark, M. Hejmo, Byzantine robust trust establishment for mobile ad hoc networks, Telecommunications Systems 35 (2007) 189–206.



Charikleia Zouridaki received the B.S. degree in Physics from Aristotle's University of Thessalonica, Greece in 2000 and the M.S. degree in Computer Engineering from George Mason University, Fairfax, VA in 2002. Currently, she is a Ph.D. Candidate in Information Technology at George Mason University, Fairfax, VA. Her research interests include network security, systems security, and communication networks. Her research focuses on security of wireless networks. Ms. Zouridaki is a Student Member of IEEE Women in Engineering (WIE). She is also a Member of Phi Beta Delta, an honor society for international scholars.

Brian L. Mark received the B.A.Sc. degree in Computer Engineering with an option in Mathematics from the University of Waterloo, Canada, in 1991 and the Ph.D. in Electrical Engineering from Princeton University, Princeton, NJ in 1995. He was a research staff member at the C&C Research Laboratories, NEC USA, Princeton, NJ from 1995 to 1999. In 1999, he was on part-time leave from NEC as a visiting researcher at Ecole Nationale Supérieure des Télécommunications in Paris, France. In 2000, he joined the Dept. of Electrical and Computer Engineering at George Mason University, where he is currently an Associate Professor. His main research interests lie broadly in the design, modeling, and analysis of communication systems, communication networks, and computer systems. He was co-recipient of the best conference paper award for IEEE Infocom'97. He received a National Science Foundation CAREER Award in 2002.

Marek Hejmo received the B.S. degree in Electrical Engineering in 1999 and the M.S. degree in Computer Engineering in 2000, both from AGH University of Science and Technology, Krakow, Poland. He completed the Ph.D. in Information Technology at George Mason University, Fairfax, VA in July 2006. Currently, he is a network security engineer with Cvent in McLean, VA. His research involves security and quality-of-service aspects of mobile ad hoc networks. Other research interests include mobile and wireless communication, ad hoc networking, performance analysis and analytical modeling.

Roshan K. Thomas received the B.Sc. degree from the University of Lagos, Nigeria and the M.S. degree in Computer Science from the University of Houston, Texas. He received the Ph.D. in Information Technology with a specialization in computer security from George Mason University, Fairfax, VA in May 1994. He is currently a Senior Principal Scientist at Sparta, Inc., and prior to that worked as Senior Scientist at McAfee Research Laboratories. He has over ten years of experience as a researcher at the Principal Investigator level in various aspects of computer security including access control models, network security, secure distributed database management and multilevel-secure object-oriented distributed computing. He is currently a co-PI on a National Science Foundation (NSF) sponsored project called SEQUOIA that is investigating the integration of security-aware quality-of-service (QoS) mechanisms in into ad-hoc wireless routing protocols. Dr. Thomas served as the co-founder of the First IEEE International Workshop on Pervasive Computing and Communication Security (PerSec 2004) and served as the PC co-chair for the second workshop (PerSec 2005).

Please cite this article in press as: C. Zouridaki et al., E-Hermes: A robust cooperative trust establishment scheme ..., Ad Hoc Netw. (2008), doi:10.1016/j.adhoc.2008.10.003