Trust and Recommendations in Mobile Ad hoc Networks

Venkat Balakrishnan, Vijay Varadharajan, Udaya Kiran Tupakula and Phillip Lucs

Information and Networked Systems Security (INSS) Research Group
Macquarie University, Sydney, Australia 2109
{venkat, vijay, udaya, phillipl}@ics.mq.edu.au

Abstract

Recently several trust and reputation models have been proposed to enhance the security of mobile ad hoc networks. In these models, recommendations are circulated by forwarding explicit messages or introducing extra message headers. Apart from incurring additional overhead, the recommendations are prone to issues such as recommender’s bias, honest-elicititation, and free-riding. In this paper, we propose a trust model to enhance the security of mobile ad hoc networks and to address the issues related to recommendations. The model uses only trusted routes for communication, and isolates malicious nodes depending on the evidence collected from direct interactions and recommendations. It deploys a novel approach for communicating recommendations such that they are free from recommender’s bias, honest-elicititation, and free-riding. Simulation results confirm the effectiveness of our model.

1. Introduction

Recently several trust and reputation based models [1-5] have been proposed to complement the secure routing models [6, 7] in mobile ad hoc networks (MANET). The trust and reputation models evaluate the trustworthiness of nodes by collecting the evidence from interactions, and recommendations. In general, the proposed models circulate recommendations by sending separate packets or adding extra headers. However, they fail to resolve the bias of a recommender, and are unable to handle the honest-elicititation or free-riding. A benign node is subject to honest-elicititation if it forwards a high recommendation for a malicious node to avoid itself from being labelled with a low recommendation by the malicious node. A malicious node may also exhibit honest-elicititation by forwarding a low recommendation for a benign node or a high recommendation for a colluding malicious node. In the case of free-riding, a node may request recommendations from other nodes but fails to reciprocate with recommendations when requested by them.

In this paper, we propose a trust model with a novel approach for communicating recommendations such that they are free from both the bias of a recommender and honest-elicititation. Furthermore, our model eliminates the necessity to forward explicit messages or additional headers for recommendations, thereby eliminating free-riding.

The paper is organized as follows. In Section 2, we outline the assumptions and the context for our model. Then in Section 3, we present our trust model and describe its operations in detail. Simulation results of our trust model are discussed in Section 4. Finally, Section 5 provides the concluding remarks.

2. Assumptions

Our trust model concentrates on enhancing the security of network layer and does not rely on any tamper proof hardware. We consider our fellowship model [9] for defending against both packet drop and flooding attacks. Dynamic Source Routing (DSR) protocol [10] is used to present the details of our trust model. We define the sequence of a successful route discovery cycle followed by data flow as a communication flow. The nodes that modify the routing information are referred as malicious nodes, and such behaviours as misbehaviours. In contrast, the behaviours that conform to the specification of the routing protocol are known as benign behaviours.

3. Trust Model

We define trust [3] as a firm belief in the competence of an entity to act as expected such that the belief is not a fixed value associated with the entity, rather it is subject to the behaviour of the entity and applies only to the given context within a defined time. It is apparent from the definition that any trust model performs the following – to make decisions depending on the trustworthiness of nodes, and to collect and maintain the evidence to evaluate the trustworthiness of nodes.
3.1. Model Operation

In the following, we demonstrate the operation of our trust model by summarizing the recommendations received for a node. Experiences with a node, while the recommended trust is based on both the direct and recommended trust corresponding to the evaluated case. The direct trust summarizes the one-to-one case. The decision for each of the cases is dependent on the corresponding trust evaluation. In turn, trust evaluations are based on both the direct and recommended trust held for one or more nodes depending on the evaluated case. The direct trust summarizes the one-to-one experiences with a node, while the recommended trust summarizes the recommendations received for a node. In the following, we demonstrate the operation of our trust model during the routing cycle.

3.2. Trust Evaluation

Our trust model assists the DSR protocol in making decisions for the following cases – to accept or reject a newly discovered route from a route discovery cycle, to record or discard a route from a forwarded packet, to choose a route from available routes for a destination, to decide whether to send a packet to a next-hop, or to forward a packet on behalf of a previous-hop. The decision for each of the cases is dependent on the direct trust evaluation. In turn, trust evaluations for the following cases – finding a route, the trust model then evaluates the route whenever it wishes to send data packets to D. On the expectation that the packet has not been modified by the previous-hop. Second, the trust for their respective next-hop is evaluated. This is performed to meet the expectation that the packet will reach the destination without modification, when the next-hop is trusted. Intermediate nodes conduct their trust evaluations for their previous-hop and next-hop nodes by using the direct and recommended trust held for them. Finally, the intermediate nodes evaluate their trust for the packet by evaluating their trust for both the source and destination nodes. This is because the intermediate nodes forward packets only for the sake of source and destination. Intermediate nodes also evaluate their trust for a route before recording the route from a forwarded packet. Similarly, D accepts the route contained in the route request only after evaluating its trust for the route.

Let us now consider the trust evaluation for – a node, a packet, and a route. Assume that the direct and recommended trust for all nodes at every node is set to the default threshold value, threshold-limit (Δ), during the initial stages of deployment. A positive trust evaluation for a node (or a packet or a route) depends on whether the trust computed for the route (or packet or route) is at least the threshold-limit (Δ).

Node ‘i’ computes its trust \( T_i-\text{Node}(t_{a+1}) \) for node ‘j’ at time ‘t_{a+1}’ by using the direct \( DT_i-\text{Node}(t_a) \) and recommended \( RT_i-\text{Node}(t_a) \) trust held for ‘j’, given in equation (1). We have set the highest priority for direct trust compared to the recommended trust as given by ‘α’ in equation (1). The reasoning rests on the intuition that personal experiences take higher precedence over the recommendations received from others.

\[
T_i-\text{Node}(t_{a+1}) = \alpha \cdot DT_i-\text{Node}(t_a) + (1-\alpha) \cdot RT_i-\text{Node}(t_a); \quad 0 < \alpha < 1; \quad t_{a+1} > t_a
\]  

Similarly, the trust \( T_i-\text{Packet}(t_{a+1}) \) for a packet ‘k’ at node ‘i’ is computed from the trust held for both the packet’s source \( T_i-\text{Node}_{\text{Src}}(t_a) \) and destination \( T_i-\text{Node}_{\text{Dest}}(t_a) \), which is given in equation (2).

\[
T_i-\text{Packet}(t_{a+1}) = \beta \cdot T_i-\text{Node}_{\text{Src}}(t_a) + (1-\beta) \cdot T_i-\text{Node}_{\text{Dest}}(t_a); \quad 0 < \beta < 1
\]

Finally, the trust \( T_i-\text{Route}(t_{a+1}) \) for a route ‘r’ at node ‘i’ is computed from the trust held for all the nodes \( T_i-\text{Node}(t_a) \) in the path, excluding node ‘i’, which is given in equation (3).

\[
T_i-\text{Route}(t_{a+1}) = \sum_{j \in \text{route and } j \neq i} \lambda_j \cdot T_i-\text{Node}(t_j); \quad \text{if } j \in \text{route and } j \neq i
\]  

Authorized licensed use limited to: MACQUARIE UNIV. Downloaded on January 12, 2009 at 17:52 from IEEE Xplore. Restrictions apply.
3.3. Direct-Trust

We define direct trust as the trustworthiness of a node depending on the evidence collected from the one-to-one interactions with the node. The evidence is collected by forwarding a packet to a next-hop and then monitoring the next-hop’s successive forwarding of the same packet. The evidence collected for the next-hop is evaluated to a positive value, \( pos(\text{packet}) \), only if the packet has been forwarded without any modification. It is noted that the magnitude of positive value is proportional to the type of packet (route request, route reply, route error or a data packet). Alternatively, if the packet is modified, then the evidence is evaluated to a negative value, \( neg(\text{packet}, \text{action}) \). Further, the malicious next-hop is excluded from the corresponding communication flow until the completion of flow. Unlike the positive value, the negative value is a function of both the type of packet and type of modification (i.e. modification of sequence number, route, source, and destination).

After evaluating the recent evidence (\( pos(\text{packet}), or \ neg(\text{packet}, \text{action}) \)), node \( i \) revises its existing direct trust for node \( j \), which is given by \( DT_i-Node_j(t_a) \) in equation (4). After the computation, the revised direct trust becomes the existing direct trust for future computation. The trust values are in the range \([-1, +1]\) as in [3]. This is because discrete trust values can only provide a small set of possible trust values, while trust in mobile ad hoc networks evolves continuously.

\[
DT_{i\rightarrow Node_j}(t_{a+1}) = \begin{cases} 
\min \{ 1, [DT_{i\rightarrow Node_j}(t_a) + pos(\text{packet})] \}, & \text{if } j \text{ is benign} \\
\max \{ -1, [DT_{i\rightarrow Node_j}(t_a) - neg(\text{packet}, \text{action})] \}, & \text{if } j \text{ is misbehaving}
\end{cases}
\]

3.4. Recommended-Trust

A recommendation can be defined as an opinion held by a recommender towards a recommended node. Recommendations are forwarded by the recommender to those nodes that are interested in the recommended node. In general, most models [1-5] communicate recommendations by forwarding explicit packets or adding extra headers. However, the models can corrupt their trust evaluations by using the recommendations received from other nodes. First, they lack the ability to determine the bias of a recommender. Second, they are short of well-developed approaches to investigate the credibility of the recommender, i.e., whether the recommender exhibits honest-elicitation and free-riding. In addition, the transmission of recommendations increases overhead and degrades the performance of the network.

3.4.1. Processing Conventional Recommendations

We believe that understanding the steps involved in forwarding and receiving a recommendation is vital in addressing the associated issues. For readability, we define a recommender as a node that is providing a recommendation for a recommended node. An interested node uses the recommendation to improve the effectiveness of its trust decisions.

First, the recommender forwards its recommendation to an interested node. The recommendation may only summarize the recommender’s one-to-one interactions with the recommended node. It may also include the summary of recommendations it had received from other nodes for the recommended node. Here, we believe that the recommendation presents the relationship between the recommender and recommended node at the current time. It is therefore feasible to deduce from a recommendation whether the recommender will forward a packet for the recommended node, given there has been no change in their relationship. This deduction applies only if the context is identical (e.g. forwarding packets) to which the recommendation refers to.

Second, the interested node evaluates the recommendation received from the recommender before revising its opinion for the recommended node. The reason for the evaluation is that the recommendation may be based on a summary of events which have not been witnessed by the interested node. Also, the recommendation reflects only the recommender’s evaluation of those events. Hence, the interested node accepts or rejects the recommendation on the grounds of its trust for the recommender. However, if it accepts the recommendation, then it scales the recommendation proportional to the level of its trust for the recommender.

3.4.2. Proposed approach for recommendations

Recall that it is feasible to deduce from an explicit recommendation, whether the recommender will forward packets on behalf of the recommended node or not. In our approach, nodes communicate recommendations by following the inverse of above-mentioned deduction process such that the recommendations are free from the associated issues. Consider the scenario from Figure 1 to understand our approach. In the scenario, node D derives an implicit recommendation from its neighbour C (recommender) for node N (recommended node) depending on whether C has forwarded packets on behalf of it previous-hop N. Node D captures the evidence for the recommendation from the route contained in a received packet. Finally, node D computes its opinion for the derived recommendation depending on its trust for the recommender, i.e., node C.
We now explain the method of validating the route contained in a packet before deriving recommendations from it. Any node deploying our trust model will forward a packet only if it trusts its previous-hop from whom it received the packet, its next-hop to whom the packet has to be forwarded, the originator of the packet (source) and the target for the packet (destination). From the above, it is evident that a node has to trust its previous-hop as one of the requirements for forwarding a received packet. Further, this applies to its previous-hop and also to all of other nodes that can be traversed backwards along the path until the originator of packet.

Consider when node X unicasts a packet to node N, containing the route S → O → X → N → C → D. Node N will forward the received packet, only if the packet is trusted. Subsequently, N derives X’s willingness to forward the packet on behalf of O, as a recommendation by X for O. Similarly, N derives O’s willingness to forward the packet on behalf of S, as a recommendation by O for S. The process of deriving recommendations terminates at S as there is no previous-hop for S.

We now explain the evaluation of the derived recommendation. Consider the recommendation derived from X (recommender) for O (recommended node) by N (interested node). Node N computes its trust for X according to equation (1). A positive or negative value is assigned for the derived recommendation depending on whether the trust for X is at least equal to the threshold-limit (Δ). This demonstrates N’s view on the recommendation derived from X for O. Note that the positive (or negative) value assigned by N for the recommendation is also identical with the positive (or negative) values assigned for recommendations derived from other nodes. This fails to present the fact that N’s trust for X need not be the same as its trust for other nodes. Hence, the positive (or negative) value representing the recommendation derived from X for O is then scaled by N’s trust for X. The scaling affirms that the recommendation is proportional to N’s trust for X. Finally, N revises its existing recommended trust for O by integrating the positive (or negative) value into its existing recommended trust for O. The same procedure is then repeated for the recommendation derived from O for S. Equation (5) summarizes the operations in which \( RT_{ij}(t_{a+1}) \) is the revised recommended trust for node ‘j’ by node ‘i’ at time ‘t_{a+1}’.

\[
RT_{ij}(t_{a+1}) = \begin{cases} 
\min\{1, [RT_{ih}(t_a) + RT_{ij}(t_a)\times pos(packet)] \} & \text{if } RT_{ih}(t_a) \geq \Delta \\
\max\{1, [RT_{ih}(t_a) - RT_{ij}(t_a)\times neg(packet)] \} & \text{if } RT_{ih}(t_a) < \Delta 
\end{cases}
\]

\( ‘i’ \) is the recommender, \( ‘j’ \) is the recommended node, and \( ‘h’ \) is the interested node.

Our analysis shows that the proposed mechanism is not foolproof both in the presence of specific secure routing model such as Secure Routing Protocol (SRP) [7] and in the absence of secure routing models. In a path, any one of the nodes prior to the node that derives the recommendations can maliciously modify the route between its position and the originator of route request. In such a situation, all the nodes that follow the malicious node in the route, derive recommendations from the modified route and eventually corrupt their existing recommended trust. Recall that the malicious node’s previous-hop would have captured the modification in its direct trust. As a result, the malicious node’s previous-hop will discard the corresponding route reply received from the malicious node. This prevents the modified route from becoming an active route for the communication flow. Hence, we defer the process of deriving recommendations to the data flow event in order to confirm that the route is active and free from modification. Once the route is active, the recommendations are derived and then evaluated but only once for the communication flow. This assures that the derived recommendations are free from corruption.

In summary, our approach to derive recommendations has several distinct advantages by eliminating -- honest-elicitation, free-riding, recommender’s bias, and the necessity to provide both authenticity and integrity for the recommendations. In addition, the proposed approach allows the node to trust itself only, as its trust ratings are neither exposed to nor influenced by other nodes. This enables a node to make its own subjective decisions. Further, the elimination of additional packets for explicit recommendations reduces the overhead and improves the performance of network.

4. **Simulation Work**

We have used the NS-2 simulator for analysing our trust model. We have evaluated our trust model in the absence of a secure routing protocol in order to understand its performance against modification attacks. The nodes which do not have the trust model are called DSR nodes. Those nodes which use the trust model are
known as trusted nodes. The nodes which perform modification attacks are called malicious nodes. The modification attacks include the addition and deletion of nodes from a route, and increasing the route request sequence number. Simulation parameters are summarized in Table 1. The Constant Bit Rate (CBR) communications among nodes are randomised such that at any one point of time there are 20 active CBR communications. When a communication terminates, a new communication is randomly generated from the set of nodes that have not been the source for a communication. The scenarios considered for simulation are:

Scenario I - The performance of trusted and DSR nodes are compared against varying proportions of malicious nodes. Maintaining the total count of nodes in the network to 100, the number of malicious nodes is increased from 0 to 100 in intervals of 10.

Scenario II - The proportion of malicious nodes is set to three fixed values, 25%, 50% and 75%. For each proportion of malicious nodes, the performance of trusted and DSR nodes is then analyzed by varying the maximum velocity \( V_{\text{max}} \) from 0m/s to 50m/s in intervals of 5m/s.

Scenario III - The impact of node density is explored in this scenario. Here instead of varying the maximum velocity, the simulation area is varied from 500m x 500m to 5000m x 5000m, in intervals of 500m x 500m.

The following performance metrics are used to evaluate the above-mentioned scenarios.

- **Packet Delivery Ratio (PDR)** is the average ratio of total number of CBR data packets received by destination to the total number of CBR packets sent by source in a communication flow.
- **Latency** is the average time taken by CBR packet to travel from source to destination in a communication flow.

### 4.1. Packet Delivery Ratio (PDR)

As shown in Figure 2(a), the PDR for trusted nodes is considerably higher than DSR nodes. The increased performance for the trusted nodes is due to the following trust decisions -- accepting packets only from a trusted previous-hop, forwarding packets only to a trusted next-hop, propagating only trusted packets, and using only trusted routes. The result confirms that the trust model is successfully able to establish valid routes.
4.2. Latency

Figure 2(d, e, and f) confirm that the latency for the trusted nodes is marginally greater than the latency of DSR nodes. The main reasons for this observation are (i) more time taken to find trusted routes as routes containing malicious nodes are discarded, (ii) the possibility of trusted routes being longer in length as they have to find paths free from malicious nodes. This is further confirmed by both DSR and trusted nodes having similar latency values in the absence of malicious nodes (Figure 2(d)). Also, we believe that there may be (iii) negligible time taken for making trust decisions at every hop, however we believe the time introduced is negligible due to the low computational and storage overheads of this trust model.

In summary, trusted nodes are capable of not only defending against malicious nodes but also efficient in not incurring additional overheads.

5. Conclusion

We have proposed a trust model with a novel approach for communicating recommendations. Compared to other trust models, our model does not incur any extra overhead in terms of either explicit messages or additional headers to circulate recommendations. Hence it not only eliminates free-riding but also addresses both honest-elicitation and recommender’s bias. Finally, we have conducted extensive simulations and confirmed the performance and efficiency of our model.

6. References