A Trust Model Robust to Slander Attacks in Ad Hoc Networks

Pedro B. Velloso[1], Rafael P. Laufer[2], Otto Carlos M. B. Duarte[3], and Guy Pujolle[1]

[1]Laboratoire d'Informatique de Paris 6 (LIP6)
UPMC - Paris, France

[2]Computer Science Department
UCLA CA, USA

[3]Grupo de Teleinformática e Automação (GTA)
UFRJ - Rio de Janeiro, RJ, Brazil

***Abstract*— Slander attacks represent a significant danger to distributed reputation systems. Malicious nodes may collude to lie about the reputation of a particular neighbor and cause serious damage to the overall trust evaluation system. This paper presents and analyzes a trust model robust to slander attacks in ad hoc networks. We provide nodes with a mechanism to build a trust relationship with its neighbors. The proposed model considers the recommendation of trustworthy neighbors and the previous experiences of the node itself. The interactions are limited to direct neighbors in order to scale on mobile networks. The results show the impact of slander attacks to our trust model. We analyze how the main parameters affect the trust evaluation process under a lying collusion attack. We show that our trust model tolerate almost 40% of liars.**

I. Introduction

The main difference between a conventional network and an ad hoc network is the lack of infrastructure. For this reason, nodes accumulate the role of router, server, and client compelling them to cooperate for the correct operation of the network. This peculiar characteristic hinders applications and protocols conceived for conventional networks to perform efficiently in ad hoc networks. Therefore, new protocols specific for this type of network have been proposed and developed. Most of the protocols and applications for ad hoc networks considers the perfect cooperation among all nodes. It is assumed that all nodes behave according to the application and protocol specifications previously defined for the network. Nevertheless, this assumption may be false, due to resource restrictions or malicious behavior. Consequently, the nodes may not behave as expected causing the network to not work properly. The assumption that nodes behave correctly can lead to unforeseen pitfalls, such as a low network efficiency, a high resource consumption, and a higher vulnerability to attacks. Therefore, a mechanism that allows a node to infer the trustworthiness of other nodes is necessary.

Providing nodes with a trust level is not only useful when nodes misbehave. In an ad hoc network there is no central entity responsible for configuring, managing, and repairing the stations. According to the paradigm of autonomic networks, a node should be capable of self-learning, self-configuring, and self-managing by means of collecting local information and exchanging information with its neighbors. Thus, it is important to communicate only with trustworthy neighbors, because the exchange of information with compromised nodes can deteriorate the autonomy of ad hoc networks. However, trust systems may suffer from slander and collusion attacks. A slander attack consists of sending false recommendations to injure the reputation of other nodes. Moreover, malicious nodes can work together to improve the effectiveness of the attack. For instance, nodes could lie about a misbehaving node to try to cover its real nature. These attacks can reduce or even ruin the performance of a distributed trust system.

Several papers propose trust models for ad hoc networks. He *et al.* [1] propose an architecture for stimulating the collaboration based on the reputation of nodes. The system is based only on the local information to evaluate the reputation of nodes. The goal is to detect and to punish nodes that do not participate in the routing process.

Pirzada and McDonald [2] propose another trust model for ad hoc networks to compute the trustworthiness of different routes. Nodes can use this information as an additional metric on routing algorithms. Although the authors present an interesting approach, the model presents several disadvantages. For instance, it is currently restricted to Dynamic Source Routing (DSR) protocol. It also relies on using promiscuous mode ignoring the energy constrains of mobile nodes. Finally, it requires each node to store information for all other nodes in the network, which is clearly non-scalable.

Virendra *et al.* [3] present an trust-based architecture that allows nodes to make decisions on establishing cryptographic keys with other nodes and forming groups of trust. Their trust self-evaluation is based on monitoring and a challenge-response system.

Theodorakopoulos and Baras [4], [5] analyze the issue of evaluating the trust level as a generalization of the shortest-path algorithm in a directed graph, where the edges correspond to the opinion that a node has about other node. They consider that nodes use just local information to establish their opinions. The opinion of each node includes the trust level and a value that represents the precision of the trust level. The main goal is to enable nodes to indirectly construct trust relationships using exclusively local information.

Sun *et al.* [6] have developed one framework capable of measuring the trust level and propagating it through the network. The goal is to secure routing and to assist intrusion detection systems. The framework also includes a defense mechanism against malicious nodes. They use a probabilistic model based on the uncertainty of a neighbor to execute one specific action and considers only local information.

We focus on providing nodes with a trust level for each direct neighbor, that is, neighbor within the radio range. The goal is to make nodes capable of gathering information to reason, learn, and make their own decisions. Different from most related works, our work improves scalability by restricting nodes to keep and exchange trust information solely with direct neighbors. We also introduce the concept of relationship maturity.

We present a trust model based on the human concept of trust. We have showed the correctness of our model in a single hop network [7]. In this paper, we analyze the robustness of our model against not only slander attacks but another lying collusion attack.

The paper is organized as follows. We present the main aspects of our trust model in Section II. Section III shows our simulation results. In Section IV we present our conclusions.

II. Trust model

The goal of the trust model is to provide nodes with a mechanism to evaluate the trust level of its direct neighbors. Our model can be divided in two distinct layers as shown is Figure 1. The Learning layer is responsible for gathering and converting information into knowledge. For instance, this layer is responsible for monitoring the behavior of each neighbor. The Trust layer then defines how to assess the trust level of each neighbor using the knowledge information provided by the Learning layer and the information exchanged with direct neighbors. Both layers can interact with all layers of the TCP/IP model. In this paper, we focus on the Trust layer and we assume an imperfect Learning layer which only perceives part of the true behavior of other nodes. The perception parameter introduced in Section III is used for this purpose.

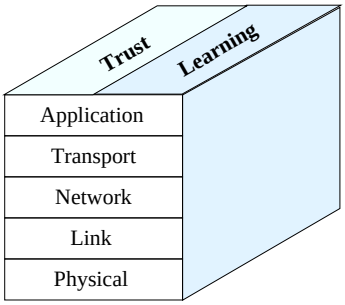


Fig. 1. Trust model.

In order to know how trustworthy a given neighbor is, each node assigns a so-called trust level for each direct neighbor. We propose a continuous representation for the trust level, ranging from 0 to 1 where 0 means the least reliable node and 1 means the most reliable node. Similar to the concept of human trust, the computation of the trust level of a given neighbor is based on previous experiences and also on the opinion of other neighbors about this specific neighbor. By previous experiences, we mean that a node keeps track of the good and bad actions taken by other neighbors. As a result, previous experiences allow a node to have a personal "opinion" about each of its neighbors. The Learning layer is the responsible for monitoring and judging other's neighbor actions. Neighbor nodes can further share their own opinions in order to improve the trust level evaluation. The transmission of a personal opinion about a specific node i is defined as a recommendation. Neighbor nodes take into account this recommendation while calculating the local trust level for node i. For that purpose, we introduce the concept of relationship maturity, which is based on the age of the relationship between two nodes. This concept allows nodes to give more importance to recommendations sent by long-term neighbors rather new neighbors. Nodes willing to consider the recommendation of other nodes use the proposed Recommendation Exchange Protocol (REP) to keep the trust level of each neighbor up to date [7]. We assume the existence of an authentication mechanism.

A. Trust level evaluation

When a node first meets a new neighbor, it must assign an initial level of trust to this neighbor. This first value depends on the network condition, level of mobility, time, and place. Afterwards, the trust level evaluation process begins with a trust recommendation request and the monitoring of the new neighbor.

We define the trust level evaluation from node a about node b as a sum of its own trust and the contribution of other nodes, in the same way as defined by Virendra *et al.* [3]. The fundamental equation is

$$T_a(b) = (1-\alpha)Q_a(b) + \alpha C_a(b), \qquad (1)$$

where α permits choosing the most relevant factor. The variable $Q_a(b)$ represents the capability of a node to evaluate the trust level of their neighbors based on its own information and $C_a(b)$ is the contribution of neighbors. In order to obtain $Q_a(b)$, we propose the following equation

$$Q_a(b) = \beta E_T + (1-\beta)T_a(b), \qquad (2)$$

where E_T represents the value obtained by the judgment of a neighbor actions, and the variable β allows choosing which factor is the more relevant at a given moment.

B. Contribution computation

The set of recommendations is called contribution ($C_a(b)$ in Equation 1). Recommendation can be obtained by sending a Trust Request (TREQ) or by receiving a Trust Advertisement (TA) message from other neighbors. TA messages are unsolicited recommendations. A node only sends a TA message when the recommendation about a particular neighbor varies more than a certain threshold value.

The contribution ($C_a(b)$) is defined as the sum of the recommendations from all nodes $i \in K_a$ about node b weighted by the trust level of node a about node i, as follows

$$C_a(b) = \frac{\sum_{i \in K_a} T_a(i) M_i(b) X_i(b)}{\sum_{j \in K_a} T_a(j) \sum_{j \in K_a} M_j(b)}. \qquad (3)$$

The group K_a defines the nodes from which recommendations will be considered. It is a subset of the neighbors of node a comprising all nodes that satisfy certain conditions. The contribution considers not only the trust level of others but also the accuracy and the relationship maturity. The accuracy of a trust level is defined by the standard deviation, similar to Theodorakopoulos and Baras [4]. The value in the trust level table of node a regarding node b is associated to a standard deviation $\sigma_a(b)$, which refers to the variations of the trust level that node a has observed about node b. We use X as a random variable with a normal distribution (N) to represent the uncertainty of the recommendation. It can be expressed as

$$X_i(b) = N(T_i(b), \sigma_i(b)). \qquad (4)$$

The recommendation of node i about node b is weighted by $M_i(b)$, Let $M_i(b)$ be defined as the maturity of the relationship between nodes i and b, measured at node i. The relationship maturity is a measure of the time that two nodes have known each other. We use the relationship maturity to give more relevance to the nodes that know the evaluated neighbor for a long time. Accordingly, we assume that the trust level of a more mature neighbor has already converged to a common value within the network and therefore its opinion should be more relevant than the opinion of a new neighbor. It is important to notice that maturity is only considered between the recommender (i) and the node that is being evaluated (b), namely, node a will never judge the opinions from neighbors that it knows longer more relevant.

Malicious nodes might try to fake trust levels for several reasons. In order to minimize this effect, each node must define a maximum relationship maturity value M_{max}, which represents an upper bound for the relationship maturity. This value is based on the average time for which a node knows its neighbors.

III. Results

In ad hoc networks, nodes might perform several actions, like sending packets, forwarding packets, responding to routing messages, among others. The set of performed actions define the node behavior. Therefore, the Learning Layer monitors the neighbor actions trying to evaluate their behavior. In our home-made simulator, each node performs good actions and/or bad actions. Nodes perform actions according to an exponential distributed variable. The kind of action that will be performed depends solely on the nature of the node. A node with a nature equals to 0.8 means that it performs eight good actions out of ten.

The nature of a node ranges from 0 to 1. Most trustworthy nodes have nature equals to 1 while nodes untrustworthy have nature equals to 0. The nature is used as a reference of the ideal global trust level that a node should receive by its neighbors. We use it here as a metric to evaluate how close the measured global trust level of a node actually gets from its nature.

Another important characteristic introduced in our simulator is the perception of a node. The perception indicates the probability of noticing a cer-

tain action. Therefore, a node with 0.6 of perception is able of noticing 60% of all the actions performed by its neighbors. This parameter simulates an interaction between the Learning Layer and the Trust Layer, since the perception and the judgment of an action is the responsibility of the Learning layer. It is worth to mention that noticing and judging an action does not imply using promiscuous mode. We believe that a node should be able to decide whether it will use promiscuous mode or not based on its own constrains and needs. Thus, nodes might decide not to use promiscuous mode at the expense of having a lower perception.

The term that considers the experiences of the own node in Equation 2 is calculated using the last i perceived actions. It implies the existence of a minimum number of actions i that a node must notice from each neighbor to be able of having an opinion about them, based on its own experience. This means that during the initial phase of first contact, nodes use just the recommendations of its neighbors to evaluate the trust level of the new one.

Our main goal in this paper is to evaluate the trust system performance under slander and collusion attacks in single-hop ad hoc networks. All results are presented with a confidence interval of 95% from a set of 100 replications. All figures present the trust evaluation of node 2 about node 1. It means that node 2 is trying to assess the trust level of node 1.

We defined the first trust assignment equal to 0.9 for every node. The first trust assignment is the level of trust that a node assigns to a neighbor, without any previous knowledge. We also chose $\alpha = \beta = perception = 0.5$. These are the standard values for the simulations. For each specific configuration, the parameters that differ from its standard values are outlined. At last, in each configuration, all nodes have nature equal to 0.9.

A. Changing behavior

In [7], we show that nodes are capable of evaluating its neighbor nature using our trust model. However, a node might change its behavior and consequently its nature during its lifetime. The behavior variation of a node occurs due to several reasons. For instance, a node may behave well at first, but after being compromised it starts to misbehave. Another possibility is a good node that experiences some energy consumption problem. Therefore, it is important for a trust model to provide nodes with the capability of identifying such behavior variations as quick as possible. Thus, in the first set of simulations we analyze the trust evaluation of a node that changes its behavior during the simulation. The scenario consists of 20 nodes with 250 m transmission range, which are randomly placed in a 150 m $\times$ 150 m area. In this particular scenario, node 1 changes its nature from 0.9 to 0.2 at 200 units of time.

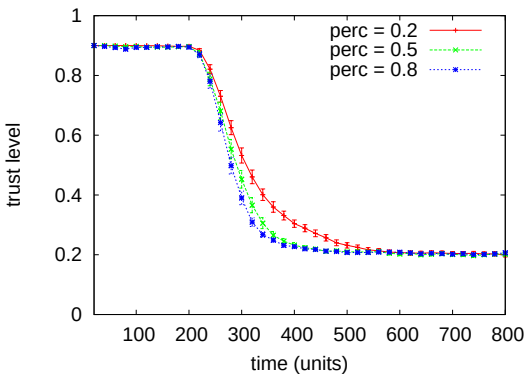
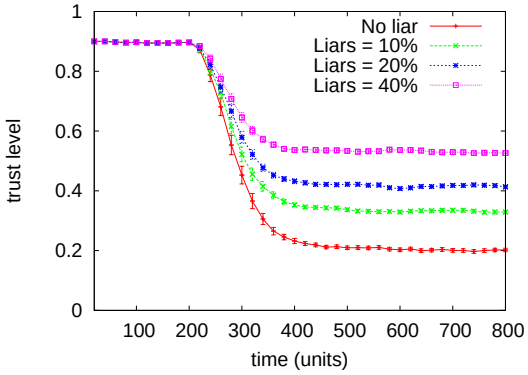


Fig. 2. Identifying behavior changes.

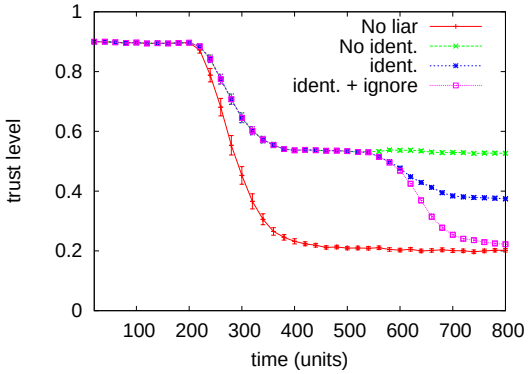
Figure 2 presents the behavior change detection according to the perception of node 2. We can notice that node 2 succeeds in all attempts to remark a change in node 1 behavior. When a node has a low perception means that it has trouble to notice its neighbor actions. This is the reason why a lower perception can slow down the trust evaluation process in the presence of behavior variations, as we can see in Figure2.

In another scenario, malicious nodes might try to cover the behavior variations of each other in order to keep a good reputation even though they have a bad behavior. Figure 3 shows a scenario where node 1 changes its nature from 0.9 to 0.2 and malicious nodes lie about node 1 trying to convince the other nodes that node 1 still have a trust level equals to 0.9. Figure 3(a) reveals the effect of a collusion attack varying the percentage of malicious nodes participating in the attack. We observe that malicious node can deteriorate the trust evaluation. However, it shows that node 2 manages to identify node 1 as a bad node, namely trust level less than 0.5, if the percentage of malicious node is smaller than 40%.

Afterwards, we propose a scenario similar to the last one, but we fixed the percentage of malicious nodes in 40%. In this scenario, we consider that nodes are capable of identifying a change in the behavior of all the malicious nodes after a certain amount of time. For instance, nodes can notice that a node is lying by comparing the recommendations it receives with its own experience during a period of time. If there is a significant discrepancy it may classify the node as malicious, and consequently, it can degrade the trust level of the detected neighbor. The results show that



(a) Varying the proportin of liars.



(b) 40% of liars.

Fig. 3. Nodes try to cover behavior changes.

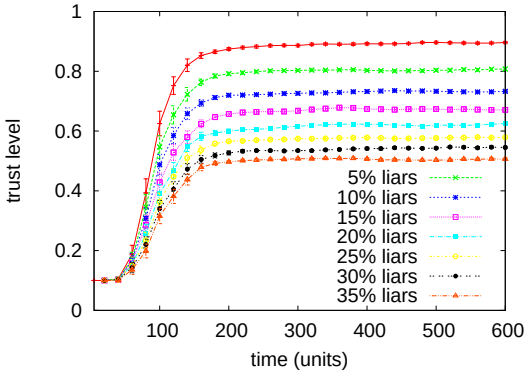


Fig. 4. Slander attack - varying the proportion of liars.

detecting liars can improve significantly the trust evaluation performance (curve "ident" Figure 3(b)) in the presence of liars. An even better solution is to detect and then to ignore completely the recommendations of malicious nodes, as shown by curve "ident + ignore" in Figure 3(b). Ignoring liars is a simple task. Node can simply ignore all recommendations of neighbors with a trust level under a certain threshold. We observe that ignoring liars can neutralize a lying collusion attack. The only damage is during the process of liar detection.

B. Slander attack

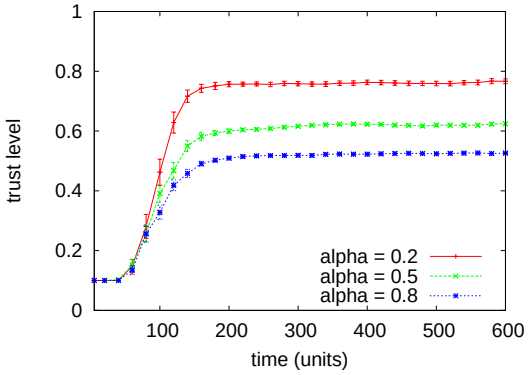
The slander attack consists of sending false recommendations to injure the reputation of a node. Malicious node can collude to improve the effect of the attack. In Figure 4, node 2 tries to evaluate the trust level of node 1 (0.9). Malicious nodes send false recommendations saying that node 1 has a trust level equals to 0.2. We vary the percentage of liars to show that node 2 can succeed in identifying node 1 as good node (Trust Level > 0.5) for a percentage of liars smaller than 40% as in the result for nodes that lie to cover behavior variations.

Figure 5, presents the result for the variation of two important parameters in our model. First, we vary $alpha$ (Figure 5(a)). The parameter $alpha$ is the one that controls the weight of recommendations and own experiences in the calculation of the trust level in Equation 1. With a higher $alpha$ the recommendations of other nodes has a higher weight on the trust level evaluation. It is clear that the more a node considers the recommendations of other nodes, the more it is vulnerable to lying attacks. Therefore, a node might have a low value for $alpha$ ($\alpha < 0.5$) in order to be more resistant to liars.

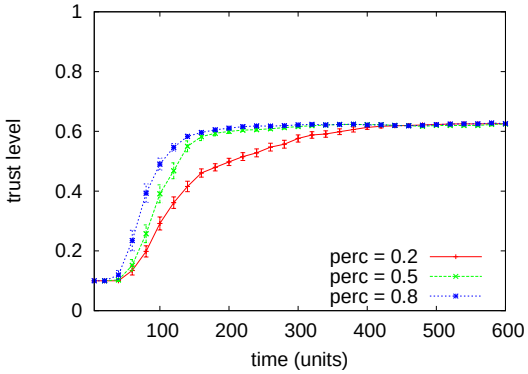
Figure 5(b) displays the impact of the perception on the slander attack. The first remark is that the perception does not impact on the trust level evaluation under a slander attack. It can be explained by the fact that the perception has influence only in the duration of the transient period and has no influence on the level achieved after convergence, in the stationary period, as shown in [7]. The transient period nodes are trying to approximate to the expected value, while in the stationary period, the trust level is almost stable, very close to the correct value.

We changed the perception of node 2 to 0.2 and the parameter alpha to 0.8 as a worst case scenario for a slander attack. Figure 6 presents the results when malicious nodes begin to lie after 200 time units so they already have a good reputation. We observe that if node 2 detects the misbehavior of the malicious nodes and ignore their recommendations (curve "lying at 200 + ident.") there is no damage to the trust evaluation process, except for the period during which node 2 has not yet notice the liars. This period depends solely on the capacity of the node in detecting a lie.

In Figure 7 we vary the duration of the detection of liars. The results show that identifying liars is an important task to avoid damage to the trust system. A fast liar detection mechanism can offer a robust trust system against slander attacks. Another possi-



(a) Varying $alpha$



(b) Varying perception

Fig. 5. Slander attack - varying trust model parameters.

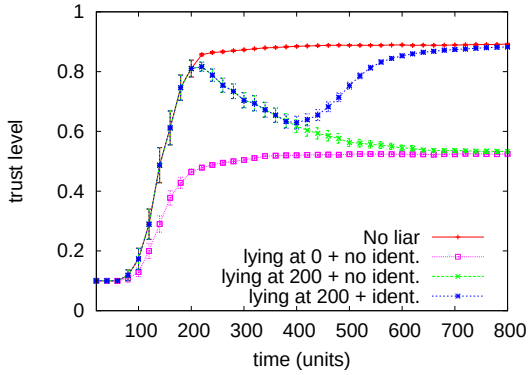


Fig. 6. Slander attack - worst case.

bility to detect liars is to compare recommendations of all neighbors. Considering that the percentage of malicious nodes is smaller than 50%, a node might assume as a liar every node that keeps sending conflicting recommendations.

IV. Conclusion

This paper presents and analyzes a robust trust model against slander attacks in ad hoc networks. We aim at building a trust relationship among nodes, confining the interactions to direct neighbors to better scale on mobile networks. We provide a mechanism for nodes to evaluate the trustworthiness of their neighbors. We analyze through simulations the performance of the proposed model in the presence of malicious nodes willing to deceive other nodes by sending false recommendations. The results show that our model tolerates almost 40% of liars. We also show that the trust system can be even more robust when nodes use a liar detection mechanism. We analyze the impact of the main parameters on the trust evaluation during a lying collusion attack.

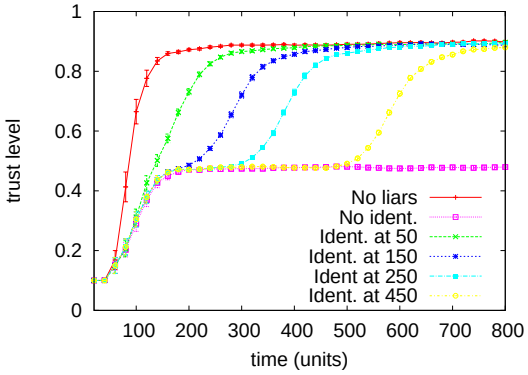


Fig. 7. Slander attack - detecting slanderer nodes.

Acknowledgment

This work has been supported by CNPq and ANR (project SARAH).

References

[1] Q. He, D. Wu, and P. Khosla, "A secure incentive architecture for ad hoc networks," in *Wireless Communications and Mobile Computing*, vol. 6, no. 3, pp. 333-346, 2006.

[2] A. A. Pirzada and C. McDonald, "Trust establishment in pure ad-hoc networks," *Wireless Personal Communications: An International Journal*, vol. 37, no. 1-2, pp. 139–168, Apr. 2006.

[3] M. Virendra, M. Jadliwala, M. Chandrasekaran, and S. Upadhyaya, "Quantifying trust in mobile ad-hoc networks," in *Proceedings of IEEE International Conference on Integration of Knowledge Intensive Multi-Agent Systems (KIMAS'05)*, (Waltham, USA), Apr. 2005.

[4] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proceedings of the ACM Workshop on Wireless Security (WiSE'04)*, (Philadelphia, USA), Oct. 2004.

[5] G. Theodorakopoulos and J. S. Baras, "On trust models and trust evaluation metrics for ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 24, no. 2, pp. 318–328, Feb. 2006.

[6] Y. Sun, Z. Han, W. Yu, and K. J. R. Liu, "A trust evaluation framework in distributed networks: Vulnerability analysis and defense against attacks," in *IEEE INFOCOM'06*, (Barcelona, Spain), Apr. 2006.

[7] P. B. Velloso, R. P. Laufer, O. C. M. B. Duarte, and G. Pujolle, "HIT: A human-inspired trust model," in *8th IFIP IEEE International Conference on Mobile and Wireless Communication Networks - MWCN'2006*, (Santiago, Chile), Aug. 2006.