Attacks on Trust Evaluation in Distributed Networks

Yan Lindsay Sun*, Zhu Han†, Wei Yu†, and K. J. Ray Liu†
*Department of Electrical and Computer Engineering
University of Rhode Island, Kingston, RI 02881
Email: yansun@ele.uri.edu
†Department of Electrical and Computer Engineering
University of Maryland, College Park, MD 20742
Emails: hanzhu, weiyu, kjrliu@glue.umd.edu

***Abstract*— Evaluation of trustworthiness of participating entities is an effective method to stimulate collaboration and improve network security in distributed networks. Similar to other security related protocols, trust evaluation is an attractive target for adversaries. Currently, the vulnerabilities of trust evaluation system have not been well understood. In this paper, we present several attacks that can undermine the accuracy of trust evaluation, and then develop defense techniques. Based on our investigation on attacks and defense, we implement a trust evaluation system in ad hoc networks for securing ad hoc routing and assisting malicious node detection. Extensive simulations are performed to illustrate various attacks, the effectiveness of the proposed defense techniques, and the overall performance of the trust evaluation system.**

I. Introduction

Currently, the networking community is working on introducing traditional security services, such as confidentiality and authentication, to distributed networks including ad hoc networks and sensor networks [1], [2]. It has been recently recognized that new tools, beyond conventional security services, need to be developed in order to stimulate cooperation in distributed networks with the presence of selfish and malicious entities [3], [4]. One of such tools is trust evaluation.

There are three primary aspects associated with evaluating trust in distributed networks. First, the ability to evaluate trust offers an incentive for good behavior. Creating an expectation that entities will "remember" one's behavior will cause network participants to act more responsibly. Second, trust evaluation provides a prediction of one's future behavior. This predication provides a means for good entities to avoid working with less trustworthy parties. Third, the results of trust evaluation can be directly applied to detect selfish and malicious entities in the network.

The research on trust evaluation has been extensively performed for a wide range of applications, including public key authentication [5], [6], electronics commerce [7], peer-to-peer networks [8], [9], ad hoc and sensor networks [10]–[13]. Currently, the design of trust evaluation systems focuses on establishing trust with high accuracy and low overhead, and on utilization of trust values to improve security and network performance. The vulnerabilities of trust evaluation, however, have not received much research attention.

Trust evaluation is an attractive target for adversaries. Besides well-known straightforward attacks such as providing dishonest recommendations [14], some sophisticated attacks can undermine the whole trust evaluation process. In this paper, we present two new attacks on trust evaluation and develop defense mechanisms. We implement a trust evaluation system in the context of ad hoc networks, and demonstrate the effects of attacks and protection schemes through simulations.

The rest of the paper is organized as follows. Section II introduces fundamental elements in trust evaluation systems, including trust definition, trust metrics, and trust models. Section III presents attacks and protection techniques for trust evaluation systems. In Section IV, the implementation of trust evaluation systems is described and simulation results are shown. The conclusion is drawn in Section VI.

II. Trust Evaluation Basis

A. Trust Concepts and Notation

Currently, there is still no clear consensus on the definition of trust in computer networks. One common interpretation of trust is belief. Briefly speaking, one entity believes that the other entity will act in a certain way [15]. Another common interpretation is probability. That is, trust is a particular level of the subjective probability with which one party assesses that another party will perform a particular action [16].

Despite the difference in definitions, trust is always established between two parties for a specific action. In particular, one party trusts the other party to perform an action. In our work, the first party is referred to as the *subject* and the second party as the *agent*. We introduce the notation $\{subject : agent, action\}$ to represent a trust relationship.

B. Trust Metrics

Trust has been evaluated by very different metrics. For example, trust is measured by linguistic descriptions in [6], discrete integers in [17], continuous value in [0, 1] in [18], a 2-tuple in $[0, 1]^2$ in [12], and a triplet in $[0, 1]^3$ in [5].

In this paper, we adopt the probability value to describe the level of trustworthiness, similar as that in [4]. Here, the probability that the agent will perform the action in the subject's point of view, denoted by $P\{subject, agent, action\}$ is used to measure trust. We adopt this metric mainly because it has a clear physical meaning. One can estimate this value based on observations. It is important to point out that the attack methods presented later in this paper do not rely on the

specific choice of trust metric. This probability based metric is used to demonstrate our ideas and implement the trust evaluation system for testing.

C. Trust Models

When the subject can directly observe the behavior of the agent, the subject can estimate the trust values based on its observations. Otherwise, the subject can establish trust in the agent based on third parties' opinions. The third parties provide recommendations by telling the subject how much they trust the agent. The later way of establishing trust is referred to as *trust propagation*.

There are two basic types of trust propagation: concatenation and multipath. In concatenation trust propagation, A and C can establish trust relationship if A has trust in B and B provides recommendation about C to A. In this case, one propagation path, A-B-C, is established. In multipath propagation, there exists multiple propagation paths. That is, A receives recommendations about C from multiple sources.

The methods for calculating trust via concatenation and multipath propagations are referred to as *trust models*. In this paper, we adopt the beta function model presented in [4] and the probability-based concatenation model in [13] with slight modification. This trust model will be briefly summarized in section IV-A.

III. Attacks and Protection

As we will show in the simulation section, trust management can effectively improve network performance. Therefore, trust management itself is an attractive target for attackers. In this section, we present the well known bad mouthing attack, identify two new attacks, and discuss defense strategies.

A. Bad Mouthing Attack

As long as recommendations are taken into consideration, malicious parties can provide dishonest recommendations [14] to frame up good parties and/or boost trust values of malicious peers. This attack is referred to as the bad mouthing attack. This attack has been discussed in many existing trust management or reputation systems [9], [14]. We summarize the defense mechanism as follows.

The defense against the bad mouthing attack relies on the usage of recommendation trust. For each entity C that A is interested in, A should maintain at least two separate trust records: $\{A : C, \text{performing action}\}$ called the action trust and $\{A : C, \text{making recommendation}\}$ called the *recommendation trust*. Only the entities who have provided good recommendations previously can earn high recommendation trust. An entity who provide dishonest recommendation will have low recommendation trust, no matter whether it performs other actions honestly or not.

Recommendation trust plays an important role in the trust evaluation. First, the trust models should be designed in such a way that the subject assign low weight to the recommendations from the nodes with lower recommendation trust. Many existing trust models have this property. Second, besides the action trust, the recommendation trust should be used in malicious entity detection process. As a result, if a node has low recommendation trust, its recommendations will have minor influence on good nodes' decision-making, and it might be detected as malicious and expelled from the network.

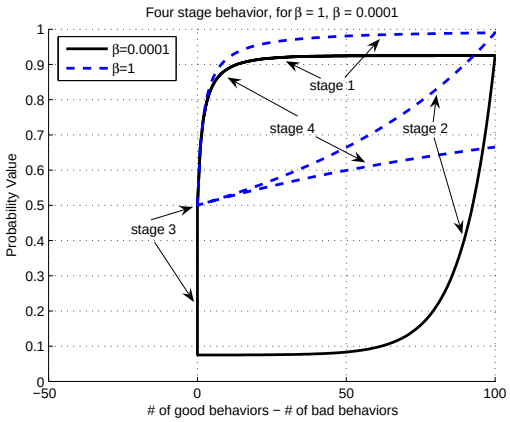

Fig. 1. Trust value changes with fixed forgetting factors

B. On-off Attack

On-off attack means that malicious entities behave well and badly alternatively, hoping that they can remain undetected while causing damage. This attack exploits the dynamic properties of trust through time-domain inconsistent behaviors. Next, we first discuss the dynamic properties of trust and then demonstrate this attack.

Trust is a dynamic event. A good entity may be compromised and turned into a malicious one, while an incompetent entity may become competent due to environmental changes. In order to track this dynamics, the observation made long time ago should not carry the same weight as that made recently. The most commonly used technique that addresses this issue is to introduce a forgetting factor. That is, performing K good actions at time t_1 is equivalent to performing $K\beta^{t_2-t_1}$ good actions at time t_2, where $\beta(0 < \beta \leq 1)$ is often referred to as the *forgetting factor*. In the existing schemes, using a fixed forgetting factor has been taken for granted. We discover, however, forgetting schemes can facilitate the on-off attack on trust management.

Let's demonstrate such an attack through a simple example. Assume an attacker behaves in the following four stages: (1) first behaves well for 100 times, (2) then behaves badly for 100 times, (3) and then stops doing anything for a while, (4) and then behaves well again. Figure 1 shows how the trust value of this attacker changes. The horizontal axis is the number of good behaviors minus the number of bad behaviors, while the vertical axis is the estimated probability value. The probability value is estimated as $\frac{S+1}{S+F+2}$, where S is the number of good behaviors and F is the number of bad behaviors. This calculation is based on the beta function model introduced in [4]. In Figure 1, the dashed line is for $\beta = 1$ and the solid line is for $\beta = 0.0001$. We observe

1. When the system does not forget, i.e. $\beta = 1$, this attacker

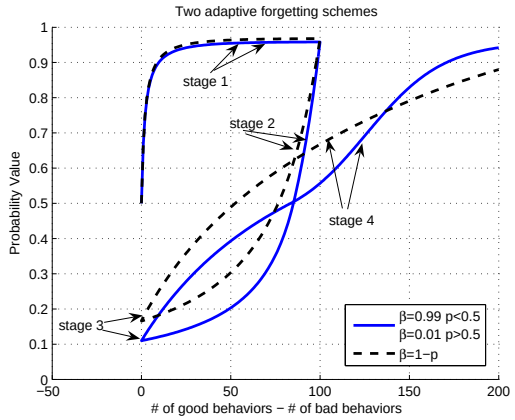

Fig. 2. Trust value changes upon entities' inconsistent behaviors with adaptive forgetting factor

has positive trust value in stage 2. That is, this attacker can have good trust values even after he has performed many bad actions. When using a large forgetting factor, the trust value may not represent the latest status of the entity. As a consequence, the malicious node could cause a large amount of damage in stage 2.

2. When using a small forgetting factor, the attacker's trust value drops rapidly after it starts behaving badly in stage 2. However, it can regain trust by simply waiting in stage 3 while the system forgets his bad behaviors quickly.

From the attacker's point of view, he can take advantage of the system no matter what forgetting factor one chooses.

To defend against the on-off attack, we propose a scheme that is inspired by a social phenomenon – while it takes long-time interaction and consistent good behaviors to build up a good reputation, only a few bad actions can ruin it. This implies that human remember bad behaviors for a longer time than they do for good behaviors. Therefore, we mimic this social phenomenon by introducing an *adaptive forgetting scheme*, where the forgetting factor is a function of the current trust value. For example, we can choose

$$\beta = 1 - p, \text{ where } p = P\{subject : agent, action\} \quad (1)$$

$$\text{or, } \beta = \beta_1 \text{ for } p \geq 0.5; \text{ and } \beta = \beta_2 \text{ for } p < 0.5, \quad (2)$$

where $0 < \beta_1 << \beta_2 \leq 1$. Figure 2 demonstrates the trust value changes when using these two adaptive forgetting schemes. The dashed line represents the case using (1), and the solid line represents the case using (2) with $\beta_1 = 0.01$ and $\beta_2 = 0.99$. Figure 2 clearly shows the advantages of the adaptive forgetting scheme. That is, the trust value can keep up with the entity's current status after the entity turns bad. And, an entity can recover its trust value after some bad behaviors, but this recovery requires many good actions.

C. Conflicting Behavior Attack

Malicious entities can impair good nodes' recommendation trust by performing differently to different peers. For example, the attackers can always behave well to one group of users and behave badly to another group of users. Thus, these two

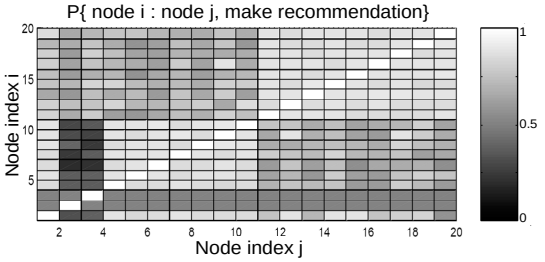

Fig. 3. Recommendation trust when malicious users attack half of good users

groups develop conflicting opinions about the malicious users. Users in the first group obtain recommendations from the other group, but those recommendations will not agree with the first group's own observations. As a consequence, the users in one group will assign low recommendation trust to the users in the other group. This attack is referred to as the conflicting behavior attack.

Figure 3 demonstrates this attack through a simple example in an ad hoc network. The system is setup as follows. In each time interval, each node randomly selects another node to transmit packets. Assume that node A selects node X. If node A does not have previous interaction with node X or the trust value $P\{A : X, \text{forward packet}\}$ is smaller than a threshold, node A asks all other nodes for recommendations about X. Then, node A asks X to forward n packets. In this example, we assume that A can observe how many packets that X has forwarded. Next, A updates the its trust record with X based on its observations and recommendation trust record with other nodes based on whether their recommendations agree with X's behavior. In this example, there are total 20 nodes. Two attackers, user 2 and 3, drop user 1, 2,$\cdots$,10's packets with packet drop ratio randomly selected between 0 and 40%, but not drop user 11,12,$\cdots$,20's packets.

In Figure 3, the element on the i^{th} row and j^{th} column represents the recommendation trust of the j^{th} user in the i^{th} user's record. The brighter the color, the higher the trust. We can see that node 1-10 will give low recommendation trust values to node 11-20, and vise versa.

D. Other Attacks

Trust evaluation systems also suffer from sybil attacks [19] and newcomer attacks [20]. In the sybil attack, a malicious node can create faked IDs that share or even take the blame, which should be given to the malicious node. In the newcomer attack, a malicious node removes its bad history by registering as a new user. The defense against the sybil attack and newcomer attack does not rely on the design of trust evaluation, but the authentication schemes. Authentication is the first line of defense that makes registering a new ID or a faked ID difficult. In this paper, we do not discuss them in depth.

IV. Simulations

A. System Description

To investigate the attacks on trust evaluation in practical systems, we implement a trust evaluation system in ad hoc

networks. The primary goal of this system is to secure ad hoc routing protocols.

In this system, we investigate trust values associated with two actions: forwarding packets and making recommendations. Briefly speaking, each node maintains trust records associated with these two actions about other nodes. When a node (source) wants to establish a route to the other node (destination), the source first tries to find multiple routes to the destination. Then the source tries to find the packet-forwarding trustworthiness of the nodes on the routes from its own trust record or through requesting recommendations. Finally the source selects the trustworthy route to transmit data. After the transmission, the source node updates the trust records based on its observation of route quality. The trust records are also used for malicious node detection.

This trust evaluation system consist of four basic building blocks: (1) Trust establishment based on observations and recommendations; (2) Trust maintenances using a proper forgetting scheme; (3) requesting/providing trust-related recommendations from/to other nodes; and (4) malicious node detection based on trust record. The first building block can be further divided into three modules: update of action trust, update of recommendation trust, and trust model. The trust evaluation system in this paper is built upon the system proposed in [13]. The main difference is the trust model and malicious node detection algorithm.

The trust model is built upon the models proposed in [4] and [13]. Let random variable P denote the probability that Y will perform the action in X's opinion. X can estimate the mean value $p_{xy} = E(P)$ and the variance value $v_{xy} = Var(P)$ based on observations. For example, if X observed that Y had performed the action successfully S times among total $(S+F)$ trails, X estimate $p_{XY} = \frac{S+1}{S+F+1}$, and $v_{XY} = \frac{(S+1)(F+1)}{(S+F+2)^2(S+F+3)}$.

In concatenation trust propagation case, let p_{AB}, v_{AB} denote the mean and variance values associated with $\{A : B, \text{make recommendation}\}$. Let p_{BC} and v_{BC} denote the mean and variance values associated with $\{B : C, action\}$. Then A calculates the mean and variance value associated with $\{A : C, action\}$ as $p_{AC} = p_{AB}p_{BC} + (1-p_{AB})(1-p_{BC})$ and $v_{AC} = p_{AB}v_{BC} + \frac{1}{12}(1-p_{AB}) + p_{AB}(1-p_{AB})(2p_{BC}-1)^2$.

In multipath trust propagation, assume A can establish trust with C through two paths. Through the first path, A calculate mean value p_1 and variance value v_1 using the concatenation model. Through the second path, A calculate mean value p_2 and v_2. Then, A calculates the final mean value (p) and variance value(v) as follows.

$$p = \frac{a}{a+b}, \quad v = \frac{ab}{(a+b)^2(a+b+1)}, \tag{3}$$

$$\text{where} \quad a = a_1 + a_2 - 1, \quad b = b_1 + b_2 - 1, \tag{4}$$

$$b_i = (1-p_i)\left(\frac{p_i(1-p_i)}{v_i} - 1\right), \tag{5}$$

$$a_i = p_i\left(\frac{p_i(1-p_i)}{v_i} - 1\right), \text{for} \quad i = 1, 2. \tag{6}$$

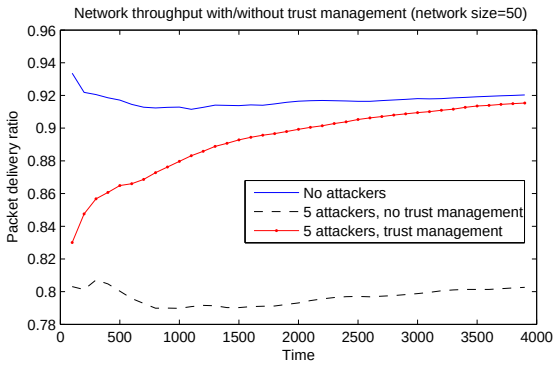

Fig. 4. Network throughput with/without trust management.

The malicious node detection algorithm is described as follows. Assume that the detection algorithm considers M trust relationships as $\{A : B, act_i\}$, for $i = 1, 2, \cdots, M$. The mean value and the variance value associated with $\{A : B, act_i\}$ is denoted by p_i and v_i, respectively. First, we convert (p_i, v_i) to (a_i, b_i) using (6) and (5). Then, we calculate $p^G_{AB} = P\{A : B, \text{be a good node}\}$ as $p^G_{AB} = \frac{a}{a+b}$, where $a = \sum_i w_i(a_i - 1) + 1$ and $b = \sum_i w_i(b_i - 1) + 1$. Here, $\{w_i\}$ is a set of weigh vectors and $w_i \leq 1$. Finally, if p^G_{AB} is smaller than a threshold, A detects B as malicious.

A simulator for ad hoc network is built, with physical layer using a fixed transmission range model, the MAC layer using IEEE 802.11 Distributed Coordination Function (DCF), and the DSR routing protocol. Approximately 50 nodes locate in a rectangular space of size 1000m by 1000m. The maximum transmission range is 300m. 50 traffic pairs are randomly generated for each simulation. For each traffic pair, the packet arrival time is modeled as a Poisson process, and the average packet inter-arrival time is 1 second. The size of each data packet is 512 bytes. Each node moves according to the random waypoint model [21] with a slight modification. A node starts at a random position, waits for a duration called the pause time that is modeled as a random variable with exponential distribution, then randomly chooses a new location and moves towards the new location with a velocity uniformly chosen between 0 and $v_{max} = 10$ meters/second. When it arrives at the new location, it waits for another random pause time and repeats the process. The average pause time is 300 seconds.

B. Effects of Trust Management

In Figure 4, three scenarios are compared: (1) baseline system that does not utilize trust management and no malicious attackers (2) baseline system with 5 attackers who randomly drop about 90% of packets passing through them; (3) the system with trust management and 5 attackers. Figure 4 shows the percentage of the packets that are successfully transmitted, which represents network throughput, as a function of time.

Three observations are made. First, network throughput can be significantly degraded by malicious attackers. Second, after using trust management, the network performance can be recovered because it enables the route selection process to avoid less trustworthy node. Third, when the simulation time increases, trust management can bring the performance close

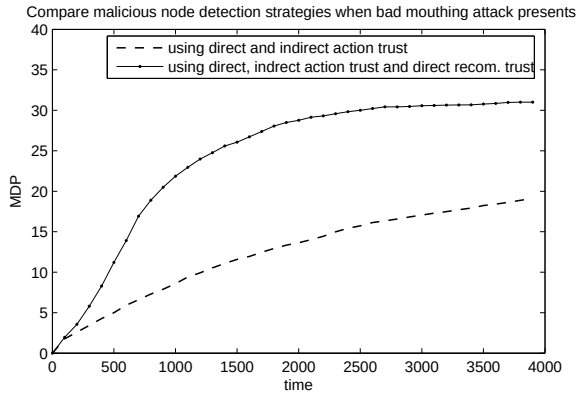

Fig. 5. Compare malicious node detection strategies when bad mouthing attack presents (50 good nodes and 5 bad nodes).

to that in the scenario where no attackers are presented, since more and more accurate trust records are built over time.

C. Bad Mouthing Attack

We introduce a metric MDP to describe the malicious node detection performance. Let D_i denote the number of good nodes who have detected that node n_i is malicious, $\mathbf{M}$ denote the set of malicious nodes, and $\mathbf{G}$ denote the set of good nodes. Then, MDP is defined as $\frac{\sum_{i:n_i \in \mathbf{M}} D_i}{|\mathbf{M}|}$, which represents the average detection rate. Similarly, we can define another metric as $\frac{\sum_{i:n_i \in \mathbf{G}} D_i}{|\mathbf{G}|}$, which describes the false alarm rate. For all simulations in this section, we choose the detection threshold such that the false alarm rate is approximately 0. Thus, we only show MDP as the performance index.

To defeat the bad mouthing attack, the best strategy is to use recommendation trust in the detection process. As illustrated in Figure 5, when using the recommendation trust in the detection process, the MDP is significantly improved, compared with the case using only packet-forwarding trust.

D. On-off Attack

For the on-off attack, we would like to compare four scenarios: (1) no on-off attack but attacking all the time; (2) with on-off attack and using forgetting factor 1 to defend; (3) with on-off attack and using forgetting factor 0.001 to defend; (4) with on-off attack and using the adaptive forgetting scheme to defend. In the last scenario, we use equation (2) in the adaptive forgetting scheme. In those experiements, when attackers are "on", they randomly choose the packet drop ratio between 40%-80%.

First, Figure 6 shows consequences of the on-off attack. With the on-off attack, the MDP values are close to 0 because attackers change behaviors when their trust values drop close to the detection threshold. Meanwhile, the network throughput is higher when attackers launch the on-off attack than that when they attack all the time.

Next, we show the tradeoff between the network throughput and the trust values of the attackers in Figure 7. The vertical axis is the average packet-forwarding trust of malicious nodes, and the horizontal axis is the network throughput. When

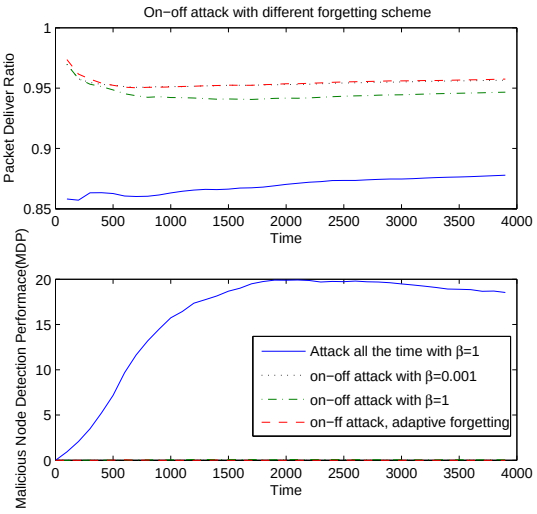

Fig. 6. The effect of on-off attack and different forgetting schemes

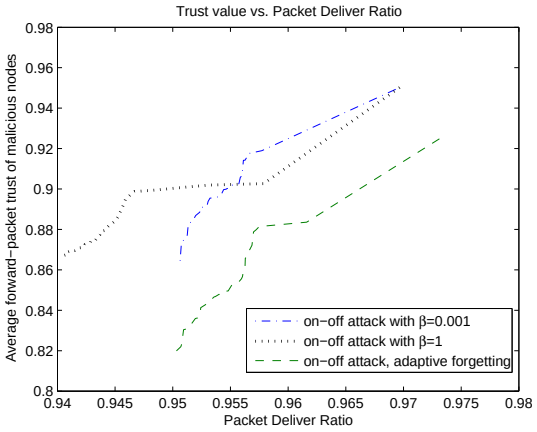

Fig. 7. Comparison between adaptive forgetting and fixed forgetting

comparing the three forgetting schemes (i.e. scenario (2)-(4)), we can see that given the same network throughput, the adaptive forgetting scheme is the best because it results in the lowest trust values for attackers.

E. Conflicting-behavior Attack

As discussed in Section III-C, the conflicting-behavior attack can deteriorate the recommendation trust of good nodes. How about the recommendation trust of bad nodes?

Assume that the attackers will drop packets for a subset of users, denoted by $\mathbf{A}$, and will not drop packets for the rest of the users, denoted by $\mathbf{B}$. The attackers have four strategies to provide recommendations to others.

(R1) providing no recommendations to $\mathbf{A}$ and honest recommendations to $\mathbf{B}$;
(R2) providing no recommendations to both $\mathbf{A}$ and $\mathbf{B}$;
(R3) providing bad recommendations to $\mathbf{A}$ and no recommendations to $\mathbf{B}$;
(R4) providing bad recommendations to $\mathbf{A}$ and honest recommendations to $\mathbf{B}$.

What is the best strategy for the attackers to make the conflicting-behavior attack more effective?

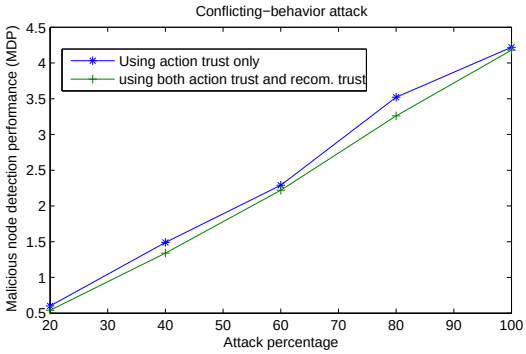

Fig. 8. Conflicting-behavior attack reduces the advantage of using recommendation trust in detection process.

We have performed extensive simulations for the above four recommendation scenarios. Due to the space limitation, the simulation curves are not included in this paper, and we only summarize the observations.

First of all, in R1 and R4, the attackers can in fact help the network performance by providing good recommendations, especially when the attack percentage is low and at the beginning of the simulation (when most good nodes have not established reliable recommendation trust with others).

In R3, malicious nodes always have much lower recommendation trust than good nodes. Thus, the conflicting behavior attack can be easily defeated as long as the threshold in the malicious node detection algorithm is properly chosen. The similar phenomenon exists in R4 when the attack percentage is high.

As a summary, if the attackers do not want to help the network by providing honest recommendations and do not want to be detected easily, the best strategy for providing recommendation is R2. Figure 8 shows the MDP values versus the percentage of users who are attacked by the malicious nodes, when R2 is adopted. The data is for the simulation time 1500. In this figure, the MDP for the detection scheme that uses packet-forwarding trust performs better than that using packet-forwarding trust and the recommendation trust. The difference between the two detection schemes in terms of MDP is not large.

In practice, when conflicting-behavior attack is suspected, one should not use recommendation trust in the detection algorithm. When it is not clear what types of attacks are launched, using recommendation trust in the malicious node detection is still a good idea because of its obvious advantages in defeating other types of attacks.

V. Conclusion

This paper presents several attack methods that can reduce the effectiveness of trust evaluation and discusses the protection schemes. In particular, we focus on bad mouthing attack, on-off attack, and conflicting-behavior attack. Simulations are performed to investigate various malicious attacks. The main observations are summarized as follows. For the bad mouthing attack, the most effective malicious node detection method is to use both packet-forwarding trust and recommendation trust. To defeat the on-off attack, the adaptive forgetting scheme developed in this paper is better than using fixed forgetting factors. From the attackers' points of view, they would not provide recommendations in order to make the conflicting-behavior attack effective. When the conflicting-behavior attack is launched, using recommendation trust in malicious node detection can reduce the detection rate. Currently, we investigate these attacks individually. In the future work, the joint effects of these attacks will be investigated.

References

[1] L. Zhou and Z. Haas, "Securing ad hoc networks," *IEEE Network Magazine*, vol. 13, no. 6, pp. 24–30, Nov./Dec. 1999.

[2] H. Chan and A. Perrig, "Security and privacy in sensor networks," *IEEE Computer*, vol. 36, no. 10, pp. 103–105, 2003.

[3] M. Blaze, J. Feigenbaum, and J. Ioannidis, "The role of trust management in distributed systems security," in *Secure Internet Programming, Springer-Verlag*, pp. 185-210, 1999.

[4] S. Ganeriwal and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," in *Proceedings of ACM Security for Ad-hoc and Sensor Networks (SASN)*, 2004.

[5] A. Jsang, "An algebra for assessing trust in certification chains," in *Proceedings of the Network and Distributed Systems Security (NDSS'99) Symposium*, 1999.

[6] M. Blaze, J. Feigenbaum, and J. Lacy, "Decentralized trust management," in *Proceedings of the 1996 IEEE Symposium on Security and Privacy*, pp. 164-173, May 1996.

[7] A. Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," in *Decision Support Systems*, 2005.

[8] P. R. Zimmermann, *The Official PGP User's Guide*, MIT Press, 1995.

[9] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina, "The eigentrust algorithm for reputation management in p2p networks," in *Proceedings of 12th International World Wide Web Conferences*, May 2003.

[10] S. Buchegger and J. L. Boudec, "Performance analysis of the confidant protocol," in *Proceedings of ACM Mobihoc*, 2002.

[11] P. Michiardi and R. Molva, "Core: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks," *Communication and Multimedia Security*, September 2002.

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

[13] Y. Sun, W. Yu, Z. Han, and K. J. Ray Liu, "Information theoretic framework of trust modeling and evaluation for ad hoc networks," *IEEE JSAC special issue on security in wireless ad hoc networks*, 2006.

[14] C. Dellarocas, "Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems," in *Proceedings of ICIS*, 2000.

[15] D. H. McKnight and N. L. Chervany, "The meanings of trust," MISRC Working Paper Series, Technical Report 94-04, arlson School of Management, University of Minnesota, 1996.

[16] D. Gambetta, "Can we trust trust?," in *in Gambetta, Diego (ed.) Trust: Making and breaking cooperative relations, electronic edition, Department of Sociology, University of Oxford*, pp. 213-237, 2000.

[17] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of 1997 New Security Paradigms Workshop, ACM Press*, pp. 48-60, 1998.

[18] U. Maurer, "Modelling a public-key infrastructure," in *Proceedings 1996 European Symposium on Research in Computer Security(ESORICS' 96), volume 1146 of Lecture Notes in Computer Science*, pp. 325-350, 1996.

[19] J. R. Douceur, "The sybil attack," in *Proceedings of First International Workshop on Peer-to-Peer systems (IPTPS'02)*, 2002.

[20] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.

[21] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks, mobile computing," *In Moible Computing, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers*, pp. 153–181, 1996.