

A Trust Evaluation Framework in Distributed Networks: Vulnerability Analysis and Defense Against Attacks

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, kjrlu@glue.umd.edu

Abstract—The performance of distributed networks depends on collaboration among distributed entities. To enhance security in distributed networks, such as ad hoc networks, it is important to evaluate the trustworthiness of participating entities since trust is the major driving force for collaboration. In this paper, we present a framework to quantitatively measure trust, model trust propagation, and defend trust evaluation systems against malicious attacks. In particular, we address the fundamental understanding of trust, quantitative trust metrics, mathematical properties of trust, dynamic properties of trust, and trust models. The attacks against trust evaluation are identified and defense techniques are developed. The proposed trust evaluation system is employed in ad hoc networks for securing ad hoc routing and assisting malicious node detection. The implementation is fully distributed. Simulations show that the proposed system can significantly improve network throughput as well as effectively detect malicious behaviors in ad hoc networks. Further, extensive simulations are performed to illustrate various attacks and the effectiveness of the proposed defense techniques.

I. INTRODUCTION

The fields of computing and communications are progressively heading towards systems of distributed entities. In the migration from traditional architectures to more distributed architectures, one of the most important challenges is security.

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]. However, it has also been recently recognized that new tools, beyond conventional security services, need to be developed in order to defend these distributed networks from misbehavior and attacks that may be launched by selfish and malicious entities [3], [4]. In fact, the very challenge of securing distributed networks comes from the *distributed* nature of these networks—there is an inherent reliance on collaboration between network participants in order to achieve the planned functionalities. Collaboration is only productive if all participants operate in an honest manner. Therefore, establishing and quantifying *trust*, which is the driving force for collaboration, is important for securing distributed networks.

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 can assist in decision-making. It provides a means for good entities to avoid working with less trustworthy parties. Malicious users, whose behavior has caused them to be recognized as having low trustworthiness, will have less ability to interfere with network operations. Third, the results of trust evaluation can be directly applied to detect selfish and malicious entities in the network.

The research on the subject of trust in computer networks has been extensively performed for a wide range of applications, including public key authentication [5]–[14], electronics commerce [15]–[17], peer-to-peer networks [18], [19], ad hoc and sensor networks [20]–[22]. However, there are still many challenges need to be addressed.

Trust definition Although definitions and classifications of trust have been borrowed from the social science literature, there is no clear consensus on the definition of trust in computer networks. Trust has been interpreted as reputation, trusting opinion, probability [23], etc.

Trust metrics Trust has been evaluated in very different ways. Some schemes employ linguistic descriptions of trust relationship, such as in PGP [18], PolicyMaker [11], distributed trust model [13], trust policy language [14], and SPKI/SDSI public-key infrastructure [12]. In some other schemes, continuous or discrete numerical values are assigned to measure the level of trustworthiness. For example, in [5], an entity’s opinion about the trustworthiness of a certificate is described by a continuous value in $[0, 1]$. In [22], a 2-tuple in $[0, 1]^2$ describes the trust opinion. In [7], the metric is a triplet in $[0, 1]^3$, where the elements in the triplet represent belief, disbelief, and uncertainty, respectively. In [13], discrete integer numbers are used.

Currently, it is very difficult to compare or validate these trust metrics because a fundamental question has not been well understood. What is the physical meaning of trust? Unlike in social networks where trust is often a subjective concept, computer networks need trust metrics to have clear physical meanings, for establishing the connection between trust metrics and observations (trust evidence) and justifying calculation/policies/rules that govern calculations performed upon trust values.

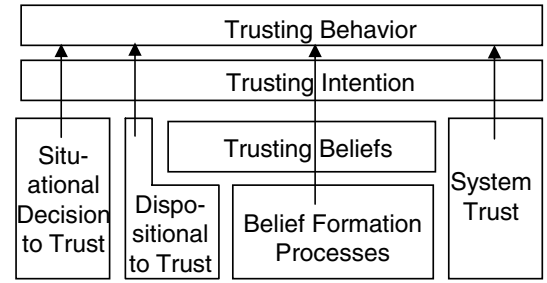
Quantitative trust models Many trust models have been developed to model trust transit through third parties. For example, the simplest method is to sum the number of positive ratings and negative ratings separately and keep a total score as the positive score minus the negative score. This method is used in eBay's reputation forum [16]. In [7], an algebra, called subjective logics, is used to assess trust values based on the triplet representation of trust. In [15], fuzzy logic provides rules for reasoning with linguistic trust metrics. In the context of the "Web of Trust", many trust models are built upon a graph where the resources/entities are nodes and trust relationships are edges, such as in [5], [6]. Then, simple mathematics, such as minimum, maximum, and weighted average, are used to calculate unknown trust values through concatenation and multipath trust propagation. In [4], [24], [25], a Bayesian model is used to take binary ratings as input and compute reputation scores by statistically updating beta probability density functions (pdf).

Although a variety of trust models are available, it is still not well understood what are the fundamental rules that trust models must follow. Without a good answer to this essential question, the design of trust models is still at the empirical stage.

Security Trust evaluation is obviously an attractive target for adversaries. Besides well-known straightforward attacks such as providing dishonest recommendations [26], some sophisticated attacks can undermine the whole trust evaluation process. In addition, providing trust recommendations may violate the privacy of individuals [27]. Currently, security and privacy issues in trust evaluation have not received enough attention.

In this paper, we address the four major challenges discussed above and develop a systematic framework for trust evaluation in distributed networks.

- We exploit the definitions of trust in the sociology, economics, political science, and psychology literature [28], [29]. By investigating correlations and differences of establishing trust in social context and that in networking, we clarify the concept of trust in distributed networks and develop trust metrics.
- We develop fundamental axioms that address the basic rules for establishing trust through a third party (concatenation propagation) and through recommendations from multiple sources (multipath propagation).
- The vulnerabilities of trust/reputation systems are extensively studied and protection strategies are proposed.



Note: Arrows indicate relationships and mediated relationships

Fig. 1. Relationship among trust constructs

Some of the vulnerabilities have not been recognized in the existing works.

- Finally, we develop a systematic framework for trust evaluation in distributed networks. To demonstrate the usage of this framework, we implement it in an ad hoc network to assist secure routing and malicious node detection. Extensive simulations are performed to demonstrate the effectiveness of the proposed trust evaluation methods and attack/antiattack schemes.

The rest of the paper is organized as follows. Section II presents understanding of trust, including trust definition, trust metrics, basic axioms for trust propagation, and trust models. Section III presents attacks and protection techniques for trust evaluation systems. In Section IV, a systematic trust management framework is introduced and applied in ad hoc networks to assist route selection and malicious node detection. Simulation results are shown in Section V, followed by the conclusion in Section VI.

II. TRUST EVALUATION FOUNDATIONS

A. Trust Concepts in Social Networks and Computer Networks

In order to understand the insightful meaning of trust, we start from the definitions of trust commonly adopted in social science [28], [29]. In [28], after examining trust definitions in 60 research articles and books, the authors identified six representative trust constructs, as illustrated in Figure 1. In social networks, trust can refer to a behavior that one person voluntarily depends on another person in a specific situation. Trust can also be an intention, that is, one party is willing to depend on the other party. For social interactions, trust intention and trust behavior are built upon four constructs: trusting belief, system trust, situational decision, and dispositional trust. Among them, the most important one is the trusting belief, that is, one believes that the other person is willing to and able to act in the other person's best interests. This belief is built upon a belief formation process. In addition, system trust means that the proper *impersonal* structures are in place to ensure successful future endeavor. Here, impersonal structures can be regulations that provide structural assurance. Dispositional trust refers to that people develop general expectation about trustworthiness of other people over the course of their lives. Situational decision trust applies to the circumstances where the benefits of trust outweigh the possible negative outcomes of the trusting behavior.

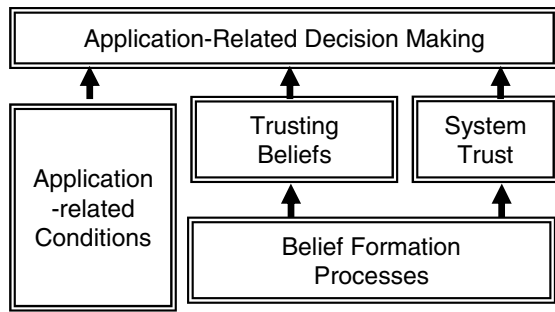


Fig. 2. Trust constructs in computer networks

The relationship among computing devices is much simpler than that among human beings. The concept of trust in computer networks does not have all six perspectives. Trust behavior, trust intension, situational decision trust and dispositional trust are not applicable to networking. Here, only *trusting belief* and *system trust*, which are built upon a *belief formation process*, are relevant to the trust concept in computer networks. In this paper, these three modules are collectively referred to as **trust management**. As illustrated in Figure 2, the outcome of trust management is provided to decision-making functions, which will make decisions based on trust evaluation as well as other application-related conditions. Further, system trust can be interpreted as a special type of belief, where an entity believes that the network will operate as it is designed. *Therefore, the most appropriate interpretation of trust in computer networks is belief.* One entity believes that the other entity will act in a certain way, or believes that the network will operate in a certain way. This is our basic understanding of trust in computer networks.

B. Notation of Trust

Trust is 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.

The concepts of subject, agent and action can have broader meanings. For example, an ad hoc mobile node trusts that the network has the capability to revoke the majority of malicious nodes. The base station trusts that the sensors around location (x,y) can successfully report explosion events. In general,

- Subject - usually represents one entity; can be a group of entities;
- Agent - one entity, a group of entities, or even the network;
- Action - an action performed (or a property possessed) by the agent.

C. Uncertainty as a Measure of Trust

Given that the trust concept in computer networks is belief, how to quantitatively evaluate the level of trust? We argue that the *uncertainty in belief* is a measure of trust. Here are three special cases.

1. When the subject believes that the agent will perform the action for sure, the subject fully trusts the agent and there is no uncertainty.
2. When the subject believes that the agent will not perform the action for sure, the subject fully distrusts the agent and there is no uncertainty either.
3. When the subject has no idea about the agent at all, there is the maximum amount of uncertainty and the subject has no trust in the agent.

Indeed, trust is built upon how certain one is about another if some actions will be carried out or not. Therefore trust metrics should describe the level of uncertainty in trust relationships.

D. Trust Metrics

How to measure uncertainty? Information theory states that entropy is a nature measure of uncertainty [30]. We would like to define a trust metric based on entropy. This metric should give trust value 1 in the first special case, -1 in the second special case, and 0 in the third special case. Let $T\{subject, agent, action\}$ denote the trust value of a trust relationship and $P\{subject, agent, action\}$ denote the probability that the agent will perform the action in the subject's point of view. In this paper, the entropy-based trust value is defined as:

$$T = \begin{cases} 1 - H(p), & \text{for } 0.5 \leq p \leq 1; \\ H(p) - 1, & \text{for } 0 \leq p < 0.5, \end{cases} \quad (1)$$

where $T = T\{subject : agent, action\}$, $p = P\{subject, agent, action\}$, $H(p) = -p \log_2(p) - (1 - p) \log_2(1 - p)$, and H is the entropy function [30]. This definition considers both trust and distrust. In general, trust value is positive when the agent is more likely to perform the action ($p > 0.5$), and is negative when the agent is more likely not to perform the action ($p < 0.5$). This definition also tells that trust value is not a linear function of the probability. This can be seen from a simple example. In the first case, let the probability increases from 0.5 to 0.509. In the second case, let the probability increase from 0.99 to 0.999. The probability value increases by the same amount in both cases, but the trust value increases by 0.00023 in the first case and 0.07 in the second case. This agrees with the intuition that the agent should gain more additional trust in the second case.

Trust is not an isolated concept. As pointed out in [28], many belief formation processes may generate belief as well as the confidence of belief. *Confidence* is an important concept because it can differentiate trust relationship established through a long-term experience and that through only a few interactions. Trust and confidence are closely related. In practice, the probability that the agent will perform the action in the subject's point of view, i.e. $p = P\{subject : agent, action\}$ is often obtained through estimation. While the belief/trust is determined by the mean value of the estimated probability, the confidence is determined by the variance of the estimation.

E. Fundamental Axioms of Trust

Trust relationship can be established through two ways: direct observations and recommendations. When direct obser-

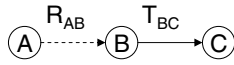


Fig. 3. Trust transit along a chain

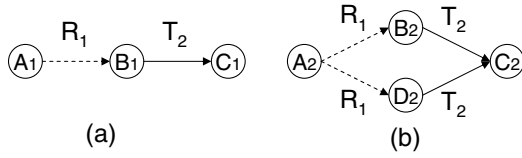


Fig. 4. Combining trust recommendations

valuations are available, the subject can estimate the probability value, and then calculate the trust value. When the subject does not have direct interaction with the agent, it can also establish trust through trust propagation. There is a need to establish the fundamental axioms that govern the basic rules of trust propagation.

Necessary Conditions of Trust Propagation

Assume that A and B have established $\{A : B, action_r\}$, and B and C have established $\{B : C, action\}$. Then, $\{A : C, action\}$ can be established if the following two conditions are satisfied.

1. $action_r$ is to make recommendation of other nodes about performing $action$.
2. The trust value of $\{A : B, action_r\}$ is positive.

The first condition is necessary because the entities that perform the action do not necessarily make correct recommendations. The second condition is necessary because untrustworthy entities' recommendation could be totally uncorrelated with the truth. The enemy's enemy is not necessarily a friend. Thus, the best strategy is not to take recommendations from untrustworthy parties.

When the above two conditions are satisfied, we recognize three axioms that are originated from the understanding of uncertainty.

Axiom 1: *Concatenation propagation of trust does not increase trust.* When the subject establishes a trust relationship with the agent through the recommendation from a third party, the trust value between the subject and the agent should not be more than the trust value between the subject and the recommender as well as the trust value between the recommender and the agent. The mathematical representation of Axiom 1 is

$$|T_{AC}| \leq \min(|R_{AB}|, |T_{BC}|), \quad (2)$$

where $T_{AC} = T\{A : C, action\}$, $R_{AB} = T\{A : B, action_r\}$ and $T_{BC} = T\{B : C, action\}$. As shown in Figure 3, the trust relationship can be represented by a directional graph, where the weight of the edge is the trust value. The style of the line represents the type of the action: dashed lines indicate making recommendations and solid lines indicate performing actions. Axiom 1 is similar to the data processing theory in information theory [30]: entropy cannot be reduced via data processing.

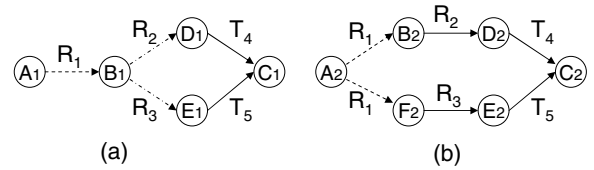


Fig. 5. Sharing entities on transit paths

Axiom 2: *Multipath propagation of trust does not reduce trust.* If the subject receives the same recommendations for the agents from multiple sources, the trust value should be no less than that in the case where the subject receives less number of recommendations.

In particular, as illustrated in Figure 4, A establishes trust with C' through one concatenation path, and A establishes trust with C through two same trust paths. Let $T_{AC} = T\{A : C, action\}$ and $T_{AC'} = T\{A : C', action\}$. The mathematical representation of Axiom 2 is

$$T_{AC} \geq T_{AC'} \geq 0, \text{ for } R_1 > 0 \text{ and } T_2 \geq 0;$$

$$T_{AC} \leq T_{AC'} \leq 0, \text{ for } R_1 > 0 \text{ and } T_2 < 0,$$

where $R_1 = T\{A : B, \text{making recommendation}\} = T\{A : D, \text{making recommendation}\}$ and $T_2 = T\{B : C, action\} = T\{D : C, action\}$. Axiom 2 states that the subject will be more certain about the agent, or at least maintain the same level of certainty if the subject obtains an extra recommendation that agrees with the subject's current opinion. Notice that Axiom 2 holds only if multiple sources generate the same recommendations. The collective combination of different recommendations is a problem in nature that can generate different trust values according to different trust models.

Axiom 3: *Trust based on multiple recommendations from a single source should not be higher than that from independent sources.*

When the trust relationship is established jointly through concatenation and multipath trust propagation, it is possible to have multiple recommendations from a single source, as shown in Figure 5 (a). Here, let $T_{AC'} = T\{A : C', action\}$ denote the trust value established in Figure 5 (a), and $T_{AC} = T\{A : C, action\}$ denote the trust value established in Figure 5 (b). For the particular case shown in Figure 5, the Axiom 3 says that

$$T_{AC} \geq T_{AC'} \geq 0, \text{ if } T_{AC'} \geq 0;$$

$$T_{AC} \leq T_{AC'} \leq 0, \text{ if } T_{AC'} < 0,$$

where R_1 , R_2 , and R_3 are all positive. Axiom 3 states that the recommendations from independent sources can reduce uncertainty more effectively than the recommendations from correlated sources.

As a summary, the above three basic Axioms address different aspects of trust relationship. Axiom 1 states the rule for concatenation trust propagation. Axiom 2 describes the rule for multipath trust propagation. Axiom 3 addresses correlation among recommendations.

F. Trust Models

The methods for calculating trust via concatenation and multipath propagations are referred to as *trust models*. Trust models should satisfy all axioms. In this section, we introduce entropy-based and probability-based trust models.

1) *Entropy-based model*: The entropy-based model takes trust values defined in (1) as the input. This model only considers trust value, but not the confidence.

For concatenation trust propagation shown in Figure 3, node B observes the behavior of node C and makes recommendation to node A as $T_{BC} = \{B : C, action\}$. Node A trusts node B with $T\{A : B, making recommendation\} = R_{AB}$. To satisfy Axiom 1, one way to calculate $T_{ABC} = T\{A : C, action\}$ is

$$T_{ABC} = R_{AB}T_{BC}. \quad (3)$$

Note that if node B has no idea about node C (i.e. $T_{BC} = 0$) or if node A has no idea about node B (i.e. $T_{AB} = 0$), the trust between A and C is zero, i.e., $T_{ABC} = 0$.

For multipath trust propagation, let $R_{AB} = T\{A : B, making recommendation\}$, $T_{BC} = T\{B : C, action\}$, $R_{AD} = T\{A : D, making recommendation\}$, $T_{DC} = T\{D : C, action\}$. Thus, A can establish trust to C through two paths: $A - B - C$ and $A - D - C$. To combine the trust established through different paths, we propose to use maximal ratio combining as:

$$T\{A : C, action\} = w_1(R_{AB}T_{BC}) + w_2(R_{AD}T_{DC}), \quad (4)$$

where

$$w_1 = \frac{R_{AB}}{R_{AB} + R_{AD}}, \text{ and } w_2 = \frac{R_{AD}}{R_{AB} + R_{AD}}. \quad (5)$$

In this model, if any path has trust value 0, this path will not affect the final result.

From (3) and (4), it is not difficult to prove that this model satisfies all Axioms.

2) *Probability-based model*: In the probability-based model, we calculate concatenation and multipath trust propagation using the probability values of trust relationship. Then, the probability values can be easily transferred back to trust values using (1). This model considers both mean and variance, i.e. trust and confidence.

Concatenation Propagation Model We first investigate the concatenation trust propagation in Figure 3. Define the following notations.

- Random variable P is the probability that C will perform the action. In A 's opinion, the trust value $T\{A : C, action\}$ is determined by $E(P)$ and the confidence is determined by $Var(P)$.
- Random variable X is binary. $X = 1$ means that B provides honest recommendations. Otherwise, $X = 0$.
- Random variable Θ is the probability that $X = 1$, i.e. $Pr(X = 1|\Theta = \theta) = \theta$ and $Pr(X = 0|\Theta = \theta) = 1 - \theta$. In A 's opinion, $P\{A : B, making recommendation\} = p_{AB} = E(\theta)$, and $Var(\theta) = \sigma_{AB}$.

- B provides recommendation about C as follows. The mean value of $P\{B : C, action\}$ is p_{BC} , and the variance value of $P\{B : C, action\}$ is σ_{BC} .

To obtain $E(P)$ and $Var(P)$, the first step is to derive the pdf of P . It is obvious that

$$f(P = p) = \int_{\theta=1}^{\theta=1} f(P = p, \Theta = \theta)d\theta, \quad (6)$$

$$f(P = p, \Theta = \theta)$$

$$= \sum_{x=0,1} f(P = p, X = x, \Theta = \theta)Pr(X = x), \quad (7)$$

$$f(P = p, X = x, \Theta = \theta)$$

$$= f(P = p|X = x, \Theta = \theta)f(X = x|\Theta = \theta)f(\Theta = \theta) \quad (8)$$

Since A 's opinion about C only depends on whether B makes honest recommendations and what B says, it is reasonable to assume that $f(P = p|X = x, \Theta = \theta) = f(P = p|X = x)$. From (8) and (7), we can see

$$f(P = p, X = x) = \theta f(P = p, X = 1)f(\Theta = \theta) + (1 - \theta)f(P = p, X = 0)f(\Theta = \theta). \quad (9)$$

From (6) and (9), we can derive that

$$f(P = p) = E(\theta)f(P = p|X = 1) + (1 - E(\theta))f(P = p|X = 0). \quad (10)$$

Using (10) and the fact that $E(\theta) = p_{AB}$, we obtain

$$E(P) = p_{AB} \cdot p_{C|X=1} + (1 - p_{AB})p_{C|X=0}, \quad (11)$$

where $p_{C|X=1} = E(P|X = 1)$ and $p_{C|X=0} = E(P|X = 0)$.

Although A does not know $p_{C|B=1}$, it is reasonable for A to assume that $p_{C|B=1} = p_{BC}$. Then, (11) becomes

$$E(P) = p_{AB} \cdot p_{BC} + (1 - p_{AB})p_{C|X=0}. \quad (12)$$

From Axiom 1, we can see that $E(P)$ should be 0.5 when p_{AB} is 0.5. By using $p_{AB} = 0.5$ and $E(P) = 0.5$ in (12), we can show that $p_{C|X=1} = (1 - p_{BC})$. Therefore, we calculate $E(P)$ as

$$E(P) = p_{AB}p_{BC} + (1 - p_{AB})(1 - p_{BC}). \quad (13)$$

Using the similar methods, $Var(P)$ is expressed as

$$\begin{aligned} Var(P) &= \int_{p=0}^{p=1} p^2 f(P = p)dp - E(P)^2 \\ &= p_{AB}\sigma_{BC} + (1 - p_{AB})\sigma_{C|X=0} \\ &\quad + p_{AB}(1 - p_{AB})(p_{BC} - p_{C|X=0})^2, \end{aligned} \quad (14)$$

where $\sigma_{C|X=0} = Var(P|X = 0)$ and $p_{C|X=0} = 1 - p_{BC}$ as in (13). The choice of $\sigma_{C|X=0}$ depends on specific application scenarios. For example, if we assume that P uniformly distributed between $[0, 1]$, we can choose $\sigma_{C|X=0}$ be the maximum possible variance, i.e. $\frac{1}{12}$. If we assume that the pdf of P is a Beta function with mean $m = p_{C|X=0}$, we can choose:

$$\sigma_{C|X=0} = \begin{cases} \frac{m(1-m)^2}{2-m} & \text{for } m \geq 0.5; \\ \frac{m^2(1-m)}{1+m} & \text{for } m < 0.5. \end{cases} \quad (15)$$

The expression in (15) is the maximum variance for a given mean m in beta distributions. As a summary, the probability-based concatenation model is expressed in (13) and (14).

Multipath Propagation Model Beta functions have been used in several schemes to address the multipath trust propagation problem [4], [24], [25]. In this section, we first briefly review the beta function model and then generalize its usage.

Assume that A can establish trust with C through two paths: $A - B - C$ and $A - D - C$. Let rec_1 represent B 's recommendation and how much A trusts B , while rec_2 represent D 's recommendation and how much A trusts D . First, when only rec_1 is available, A uses Bayesian model and obtain:

$$f(P = p | rec_1) = \frac{Pr(rec_1|P = p) \cdot f_0(P = p)}{\int Pr(rec_1|P = p) \cdot f_0(P = p)dp}, \quad (16)$$

where $f_0(P = p)$ is the prior knowledge of P . When A does not have previous knowledge of P , we assume $f_0(P = p)$ is a uniform distribution between $[0, 1]$. Thus,

$$f(P = p | rec_1) = \frac{Pr(rec_1|P = p)}{\int Pr(rec_1|P = p)dp} \quad (17)$$

Next, A obtains more information about C through the second path as rec_2 . We use the Bayesian model again and replace the prior knowledge with $f(P = p|rec_1)$ as:

$$\begin{aligned} f(P = p|rec_2, rec_1) \\ = \frac{Pr(rec_2|P = p) \cdot f(P = p|rec_1)}{\int Pr(rec_2|P = p) \cdot f(P = p|rec_1)dp} \end{aligned} \quad (18)$$

If we assume that $Pr(rec_1|P = p)$ and $Pr(rec_2|P = p)$ are beta functions, i.e.

$$Pr(rec_1|P = p) = B(\alpha_1, \beta_1), \quad (19)$$

$$Pr(rec_2|P = p) = B(\alpha_2, \beta_2), \quad (20)$$

then, it can be proved that (18) is also a beta function as $B(\alpha_1 + \alpha_2 - 1, \beta_1 + \beta_2 - 1)$. The beta distribution is

$$B(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1} (1-p)^{\beta-1}. \quad (21)$$

The Beta function model is often used in the scenarios where the subject has collected binary opinions/observation about the agent [4], [25]. For example, entity A receives total S positive feedback and F negative feedback about entity C . In another example, entity A made an observation that C had performed the action successfully S times among total $S + F$ trails. In these cases, the probability $Pr(observation|P = p)$ is approximately $B(S + 1, F + 1)$.

Next, we generalize the usage of the beta function model to non-binary opinions/observation cases. It is known that the Beta distribution $B(\alpha, \beta)$ has mean and variance as

$$m = \frac{\alpha}{\alpha + \beta}; \quad v = \frac{\alpha\beta}{(\alpha + \beta)^2(\alpha + \beta + 1)}. \quad (22)$$

Thus, the parameter α and β are determined from mean and variance as:

$$\alpha = m \left(\frac{m(1-m)}{v} - 1 \right); \quad \beta = (1-m) \left(\frac{m(1-m)}{v} - 1 \right). \quad (23)$$

In the multipath trust propagation case, let A establish trust and confidence represented by the mean value m_1 and the variance value v_1 through the first path. Through the second path, A establishes trust and confidence represented by the mean value m_2 and the variance value v_2 . Using equation (23), (m_1, v_1) is converted to (α_1, β_1) and (m_2, v_2) is converted to (α_2, β_2) . Then, a pair of new parameter (α, β) is calculated as $\alpha = \alpha_1 + \alpha_2 - 1$ and $\beta = \beta_1 + \beta_2 - 1$. After combining the two paths, the new mean value and variance value should be calculated from (α, β) using equation (22).

III. ATTACKS AND PROTECTION

As we will show in the simulation section, trust management can effectively improve network performance and detect malicious entities. Therefore, trust management itself is an attractive target for attackers. Besides some well known attacks, such as bad mouthing attack, we will identify new attacks and develop defense methods in this section.

A. Bad Mouthing Attack

As long as recommendations are taken into consideration, malicious parties can provide dishonest recommendations [26] to frame up good parties and/or boost trust values of malicious peers. This attack, referred to as the bad mouthing attack, is the most straightforward attack and has been discussed in many existing trust management or reputation systems.

In our work, the defense against the bad mouthing attack has three perspectives. First, the action trust and the recommendation trust records are maintained separately. Only the entities who have provided good recommendations previously can earn high recommendation trust. Second, recommendation trust plays an important role in the trust propagation process. The necessary conditions of trust propagation state that only the recommendations from the entities with positive trust values can propagate. In addition, the three fundamental axioms limit the recommendation power of the entities with low recommendation trust. Third, besides the action trust, the recommendation trust is treated as an additional dimension in the 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 can be detected as malicious and expelled from the network. The consequences of the bad mouthing attack and the effectiveness of the defense strategy will be demonstrated in Section V.

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.

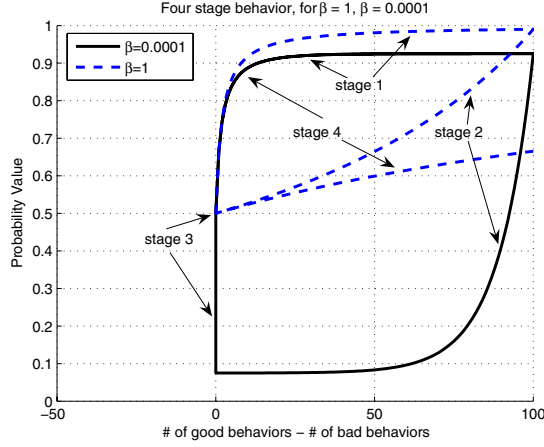


Fig. 6. Trust value changes upon entities' inconsistent behaviors with fixed forgetting factor

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 wireless networks, for example, a mobile node may experience bad channel condition at a certain location and has low trust value associated with forwarding packets. After moving to a new location where the channel condition is good, some mechanisms should be in place to recover its trust value.

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 factor 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 6 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 Section II-F.2. In Figure 6, the dashed line is for $\beta = 1$ and the solid line is for $\beta = 0.0001$. Then, we observe

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

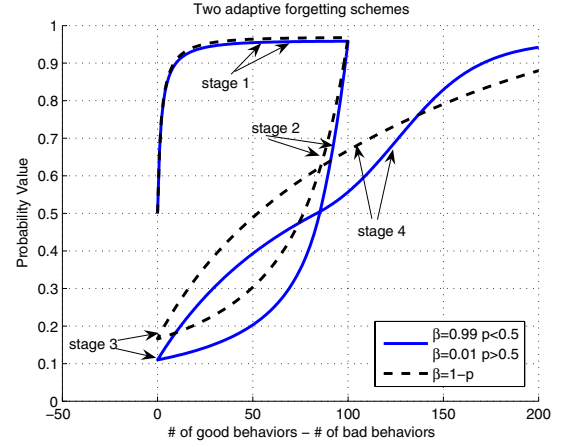


Fig. 7. Trust value changes upon entities' inconsistent behaviors with adaptive forgetting factor

a large amount of damage in a stage that is similar to 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 will forget his bad behaviors quickly.

From the attackers' point of view, he can take advantage of the system one way or another, 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*. Instead of using a fixed forgetting factor, β is a function of the current trust value. For example, we can choose

$$\beta = 1 - p, \text{ where } p = P\{\text{subject} : \text{agent}, \text{action}\} \quad (24)$$

$$\text{or, } \beta = \beta_1 \text{ for } p \geq 0.5; \text{ and } \beta = \beta_2 \text{ for } p < 0.5, \quad (25)$$

where $0 < \beta_1 < \beta_2 \leq 1$. Figure 7 demonstrates the trust value changes when using these two adaptive forgetting schemes. The dashed line represents the case using equation (24), and the solid line represents the case using equation (25) with $\beta_1 = 0.01$ and $\beta_2 = 0.99$. Figure 7 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

While an attacker can behave inconsistently in the time domain, he can also behave inconsistently in the user domain. In particular, malicious entities can impair good nodes' recommendation trust by performing differently to different peers. This attack is referred to as the conflicting behavior attack.

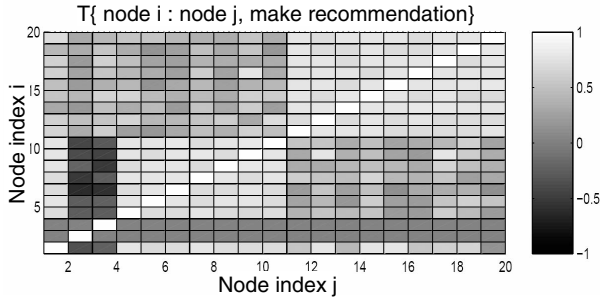


Fig. 8. Recommendation trust when malicious users attack half of good users

For example, the attackers can always behave well to one group of users and behave badly to another group of users. Thus, these two 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.

Figure 8 demonstrates this attack through a simple example in an ad hoc network. The system is setup as follows. In each time interval, which is n time units long, 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 $T\{A : X, \text{forward packet}\}$ is smaller than a threshold, node A ask 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. The detailed trust updating procedure will be described in Section IV. In this example, there are total 20 nodes. If a malicious node decides to attack node A , it drops the packets from A with packet drop ratio randomly selected between 0 and 40%. Two attackers, user 2 and 3, launch the conflicting behavior attack by dropping user 1, 2, ..., 10's packets but not dropping user 11, 12, ..., 20's packets.

In Figure 8, 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 vice versa.

D. Sybil Attack and Newcomer Attack

If a malicious node can create several faked IDs, the trust management system suffers from the *sybil attack* [31], [32]. The faked IDs can share or even take the blame, which should be given to the malicious node.

Here is an example of the sybil attack. In an ad hoc network, node A sends packets to node D through a path $A-B-C-D$. With the sybil attack, B creates a faked ID B' and makes the route look like $A-B-B'-C-D$ from A, C, D 's point. Node B can achieve this by manipulating route discovery messages, communicating with A using ID B and communicating with C using ID B' . When packets are dropped by B , B could make B' take the blame if B is ever suspected for dropping

packets. Obviously, B can also created multiple faked IDs.

If a malicious node can easily register as a new user, the trust management suffers from the *newcomer attack* [33]. Here, malicious nodes can easily remove their bad history by registering as a new user. New comer attack can significantly reduce the effectiveness of trust management.

The defense to the sybil attack and newcomer attack does not rely on the design of trust management, 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 simply point out these two attacks and will not discuss them in depth.

IV. TRUST MANAGEMENT SYSTEMS AND ITS APPLICATIONS IN AD HOC NETWORKS

A. Design of Trust Management Systems

In the current literature, many works use heuristic trust metrics to address one or a few perspectives of trust management for specific applications. There are few works focusing on establishing generic trust models [34] or providing a complete picture of trust management through a survey [17]. However, the existing works do not well address two important perspectives of trust management in distributed computer networks. The first is the networking specific elements such as how to request and obtain recommendations, and the second is attacks and protection mechanisms.

In this paper, we design a comprehensive framework of trust management for distributed networks, as illustrated in Figure 9. This framework contains five basic building blocks. *Trust record* is constructed through the *trust establishment* process, which builds direct trust values from observations and indirect trust values from recommendations, and updated by the *record maintenance* process, which assigns initial trust values and addresses dynamic properties of trust. *Trust requests management* serves as the interface between applications that request trust values and trust management. It also handles the requests for trust recommendations. In addition, *malicious node detection* is performed based on trust record and its output also affects some entries in the trust record. This framework can be used in a variety of applications, such as ad hoc networks, peer-to-peer networks, and sensor networks. To demonstrate its usage, we present the implementation of such a framework in mobile ad hoc networks.

B. Applications in Ad hoc Networks

In ad hoc networks, securing routing protocols is one of the fundamental challenges [35]–[37]. While many secure routing schemes focus on preventing attackers from entering the network through secure key distribution/authentication and secure neighbor discovery, such as in [36], [38], trust management can guard routing even if malicious nodes have gained access to the network. In this section, we demonstrate the usage of trust management in ad hoc network to secure routing protocols.

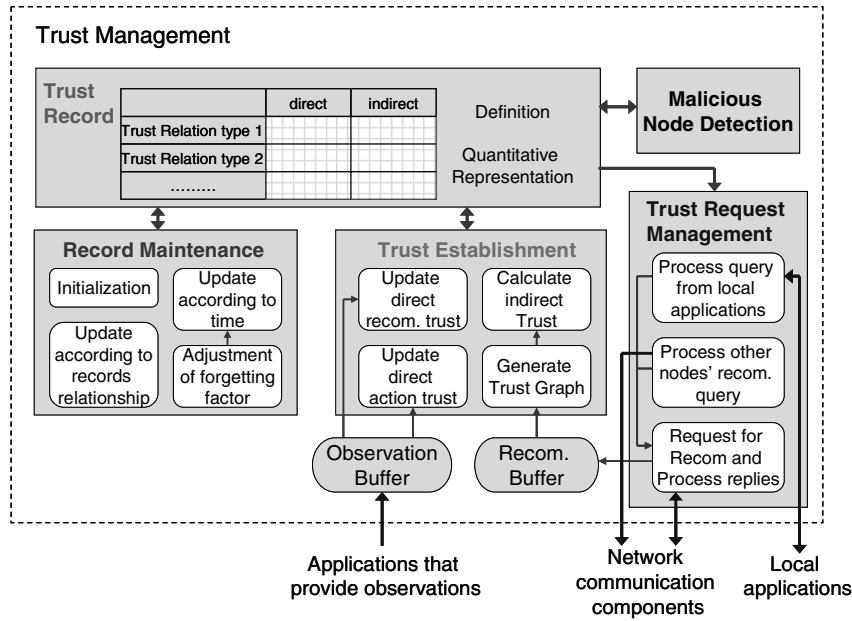


Fig. 9. Trust management system for distributed computer networks

For ad hoc routing, we investigate the trust values associated with two actions: forwarding packets and making recommendations. Briefly speaking, each node maintains its trust record associated with these two actions. 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. All above is achieved in a distributed manner.

1) *Obtaining trust recommendations*: Requiring trust recommendation in ad hoc networks often occurs in the circumstance where communication channels between arbitrary entities are not available. In this section, we briefly introduce the procedures for requesting trust recommendations and processing trust recommendation requests.

We assume that node A wants to establish trust relationships with a set of nodes $B = \{B_1, B_2, \dots\}$ about action act , and A does not have valid trust record with $\{B_i, \forall i\}$. Node A first checks its trust record and selects a set of nodes, denoted by \hat{Z} , that have the recommendation trust values larger than a threshold. Although A only needs recommendations from \hat{Z} to calculate the trust value of B , A may ask for recommendations from a larger set of nodes, denoted by Z , for two reasons. First, node A does not necessarily want to reveal the information about whom it trusts because the malicious nodes may take advantage of this information. Second, if node A establishes trust with B through direct interaction later, node A can use the recommendations it collects previously to update the recommendation trust of the nodes in Z . Thus, Z should contain not only the nodes in \hat{Z} , but also the nodes with which A wants to update/establish recommendation trust. Next, node

A sends a trust recommendation request (TRR) message to its neighbors that in node A 's transmission range. The TRR message should contain the IDs of nodes in set B and in set Z . In order to reduce overhead, the TRR message also contains the maximal length of trust transit chains, denoted by $Max_transit$, and time-to-live (TTL). Node A waits time TTL for replies. In addition, *transmit-path* is used to record delivery history of the TRR message such that the nodes who receive the TRR message can send their recommendations back to A .

Upon receiving an unexpired TRR message, the nodes that are not in Z simply forward the TRR message to their neighbors; the nodes in Z either send trust values back to A or ask their trusted recommenders for further recommendations. In addition, the nodes in Z may not respond to the TRR message if they do not want to reveal their trust records to A when, for example, they believe that A is malicious.

In particular, suppose node X is in Z . When X receives an unexpired TRR message, if X has the trust relationship with some of $\{B_i\}$'s, X sends its recommendation back to A . If X does not have trust relationship with some of $\{B_i\}$'s, X generates a new TRR message by replacing Z with the recommenders trusted by X and reducing the value of $Max_transit$ by one. If $Max_transit > 0$, the revised TRR message is sent to X 's neighbors. X also sends A corresponding recommendation trust values needed for A to establish trust propagation paths. If the original TRR message has not expired, X will also forward the original TRR message to its neighbors. By doing so, the trust concatenations can be constructed.

The major overhead of requesting trust recommendations comes from transmitting TRR messages in the network, which increases exponentially with $Max_transit$. Fortunately, $Max_transit$ should be a small number due to Axiom 1, which implies that only short trust transit chains are useful.

2) *Trust record maintenance and update*: In this study, the trust relationship $\{A : C, \text{forward packet}\}$ is established based on whether C forwarded packets for A . Assume that A asked C to forward N packets and C actually forwarded k packets. Node A will calculate $P\{A : C, \text{forward packet}\} = \frac{k+1}{N+2}$. The observation of k and N values are made through a light-weight self-evaluation mechanism, which allows the source node to collect packet forwarding statistics and to validate the statistics through consistence check. More details of this mechanism is presented in [39]. In addition, before any interaction takes place, A sets the initial trust values using $k = 0$ and $N = 0$.

Next, we present the procedure of updating trust records. Assume that node A would like to ask C to transmit packets, while A does not have trust relationship with node C .

Before data transmission

- Node A receives the recommendation from node B , and node B says that $T\{B : C, \text{forward packet}\} = T_{BC}$. And A has established $\{A : B, \text{make recommendation}\}$ previously. Then, A calculates $T_{AC}^r = T\{A : C, \text{forward packet}\}$ based on trust propagation models.

After data transmission

- Node A observes that C forwards k packets out of total N packets.
- A calculates $T\{A : C, \text{forward packet}\}$ based on observations as T_{AC}^a , and updates its trust record.
- If $|T_{AC}^a - T_{AC}^r| \leq \text{threshold}$, node A believes that B has made one good recommendation. Otherwise, node A believes that B has made one bad recommendation. Then, A can update the recommendation trust of B accordingly.

3) Some implementation details:

- Route discovery: A performs on-demand routing to find several possible routes to destination D .
- Route selection: Among all possible routes, node A would like to choose a route that has the best quality. Let $\{n_i, \forall i\}$ represent the nodes on a particular route R . Let p_i represent $P\{A : n_i, \text{forward packet}\}$, where A is the source. The quality of route R is calculated as $\prod_i p_i$.
- Malicious Node Detection: Assume that the malicious node detection algorithm considers M trust relationships as $\{A : B, act_i\}$, for $i = 1, 2, \dots, M$. The mean value and the variance value associated with $\{A : B, act_i\}$ is denoted by m_i and v_i , respectively. First, we convert (m_i, v_i) to (α_i, β_i) using (23). Then, we calculate $p_{AB}^G = P\{A : B, \text{be a good node}\}$ as $p_{AB}^G = \frac{\alpha}{\alpha + \beta}$, where $\alpha = \sum_i w_i(\alpha_i - 1) + 1$ and $\beta = \sum_i w_i(\beta_i - 1) + 1$. Here, $\{w_i\}$ is a set of weigh vectors and $w_i \leq 1$. Finally, if p_{AB}^G is smaller than a threshold, A detects B as malicious.

V. SIMULATIONS

An event-driven simulator is built to simulate mobile ad hoc networks. The physical layer uses a fixed transmission range model, where two nodes can directly communicate with each other only if they are within a certain transmission range. The MAC layer protocol simulates the IEEE 802.11 Distributed Coordination Function (DCF) [40]. DSR is used

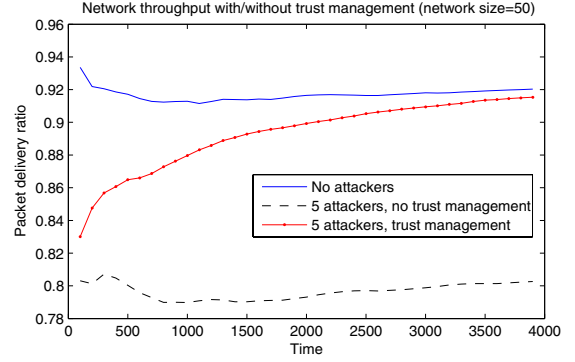


Fig. 10. Network throughput with/without trust management.

as the underlying routing protocol. We use a rectangular space of size 1000m by 1000m. The network size is about 50 nodes, and the maximum transmission range is 300m. There are 50 traffic pairs 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 after encryption is 512 bytes. Among all the ROUTE REQUESTs with the same ID received by node A , node A will only broadcast the first request if it is not the destination, and will send back at most 5 ROUTE REPLYs if it is the destination. The maximum number of hops on a route is restricted to be 10. Each node moves randomly according to the random waypoint model [41] with a slight modification. A node starts at a random position, waits for a duration called the pause time 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.

In this section, we first show the advantages of trust management in improving network throughput and malicious detection, and then demonstrate the effects of several attack/anti-attack methods presented in Section III.

A. Effects of Trust Management

In Figure 10, 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. Here, we use the probability-based trust model. Figure 10 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 to that in the scenario where no attackers are presented, since more and more accurate trust records are built over time.

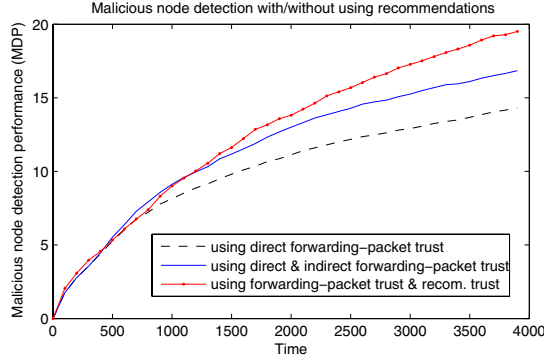


Fig. 11. The effectiveness of malicious node detection with/without recommendations.

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.

Figure 11 shows the MDP for three cases. In case 1, only direct packet-forwarding trust information is used to detect malicious nodes. In case 2, both direct and indirect packet-forwarding trust information is used to detect malicious nodes. In case 3, direct and indirect packet-forwarding trust and direct recommendation trust are used. Recall that direct trust records are built upon the observations, while indirect trust records are built upon the recommendations. As we expected, the detection rate is higher when indirect information and recommendation trust information is used. This means that the recommendation mechanism improves the performance of malicious node detection.

B. Bad Mouthing Attack

The influence of the bad mouthing attack is demonstrated in Figure 12, which shows the network throughput when attackers only launch the gray hole attack (i.e. dropping packets) and when attackers launch both the gray hole and bad mouthing attack. Here, both direct and indirect packet-forwarding trust is used in the route selection process. We can see that the bad mouthing attack leads to a performance drop since indirect trust information can be inaccurate. However, this performance drop is small because our trust management system already has defense mechanisms embedded, as discussed in Section III-A.

To defeat the bad mouthing attack, the best strategy is to use recommendation trust in the detection process. As illustrated in Figure 13, when using the direct recommendation trust in the detection process, the MDP is significantly improved, compared with the case using only packet-forwarding trust.

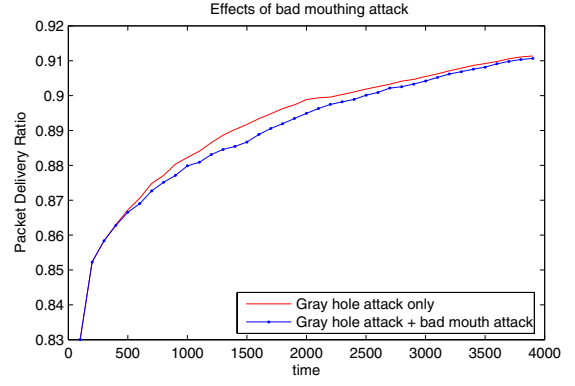


Fig. 12. The effects of bad mouthing attack when route selection uses both direct and indirect packet-forwarding trust information. (50 good nodes, 5 bad nodes)

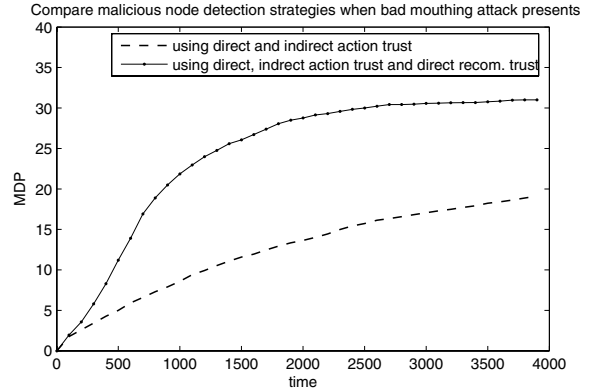


Fig. 13. Compare malicious node detection strategies when bad mouthing attack presents (50 good nodes and 5 bad nodes).

C. 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 (25) in the adaptive forgetting scheme. In those experiments, when attackers are “on”, they randomly choose the packet drop ratio between 40%-80%.

First, Figure 14 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 15. The vertical axis is the average packet-forwarding trust of malicious nodes, and the horizontal axis is the network throughput. When 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.

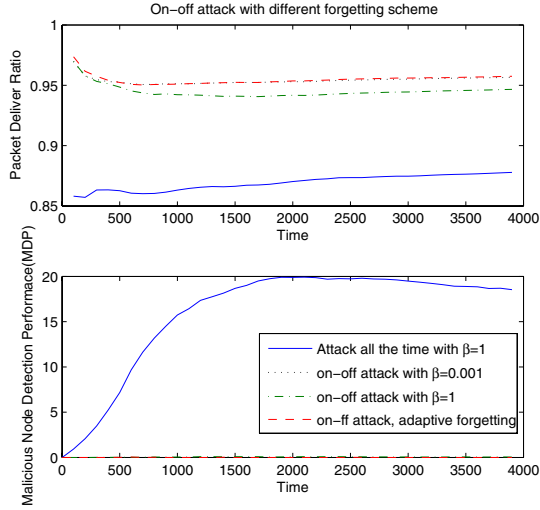


Fig. 14. The effect of on-off attack and different forgetting schemes

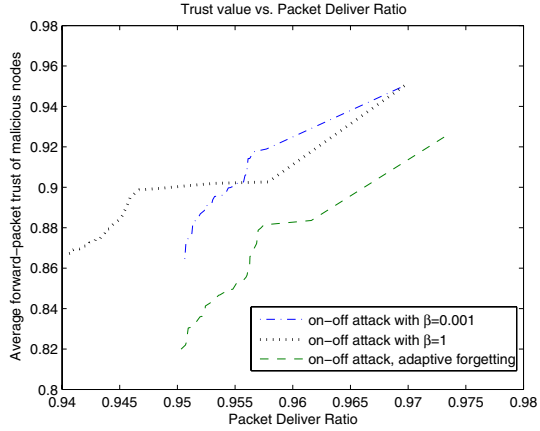


Fig. 15. Comparison between adaptive forgetting and fixed forgetting

D. 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?

- The attackers have four strategies to provide recommendations to others. Assume that the attackers will drop packets for a subset of users, denoted by **A**, and will not drop packets for the rest of the users, denoted by **B**. The attackers can provide
- R1. no recommendations to subgroup A, and honest recommendations to subgroup B;
 - R2. no recommendations to subgroup A, and no recommendations to subgroup B;
 - R3. bad recommendations to subgroup A, and no recommendations to subgroup B;
 - R4. bad recommendations to subgroup A, and honest recommendations to subgroup B.

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

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

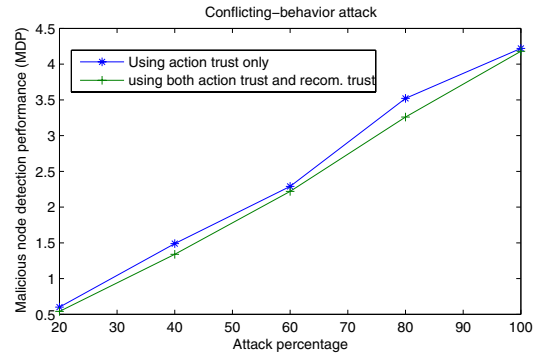


Fig. 16. Conflicting-behavior attack reduces the advantage of using recommendation trust in detection process.

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 R1, malicious nodes usually have higher recommendation trust than good nodes. Thus, it is harmful to use the recommendation trust in the malicious node detection algorithm. The similar phenomenon exists in R4 when the attack percentage is low.

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 with high attack percentage.

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 16 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 direct and indirect packet-forwarding trust performs better than that using packet-forwarding trust and the recommendation trust. In addition, the difference between the two detection schemes in terms of MDP is not very 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.

VI. CONCLUSION

This paper presents a framework for trust evaluation in distributed networks. We address the concept of trust in computer networks, develop trust metrics with clear physical meanings, develop fundamental axioms of the mathematical properties of trust, and build trust models that govern trust propagation through third parties. Further, we present attack methods that can reduce the effectiveness of trust evaluation

and discuss the protection schemes. Then, a systemic trust management system is designed, with the specific consideration of distributed implementation. In this work, the usage of the proposed framework is demonstrated in ad hoc networks to assist route selection and malicious node detection.

The simulation results show that the proposed trust evaluation system can improve network throughput as well as help malicious node detection. Simulations are also 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] 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.
- [6] M. K. Reiter and S. G. Stubblebine, "Resilient authentication using path independence," *IEEE Transactions on Computers*, vol. 47, no. 12, pp. 1351–1362, December 1998.
- [7] A. Jsang, "An algebra for assessing trust in certification chains," in *Proceedings of the Network and Distributed Systems Security (NDSS'99) Symposium*, 1999.
- [8] R. Levien and A. Aiken, "Attack-resistant trust metrics for public key certification," in *Proceedings of the 7th USENIX Security Symposium*, pp. 229–242, January 1998.
- [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] R. Guha, R. Kumar, P. Raghavan, and A.T. Propagation, "Propagation of trust and distrust," in *Proceedings of International World Wide Web Conference*, 2004.
- [11] 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.
- [12] D. Clarke, J.-E. Elien, C. Ellison, M. Fredette, A. Morcos, and R. L. Rivest, "Certificate chain discovery in spki/sdsi," *Journal of Computer Security*, vol. 9, no. 4, pp. 285–322, 2001.
- [13] A. Abdul-Rahman and S. Hailes, "A distributed trust model," in *Proceedings of 1997 New Security Paradigms Workshop*, ACM Press, pp. 48–60, 1998.
- [14] A. Herzberg, Y. Mass, J. Michaeli, D. Naor, and Y. Ravid, "Access control meets public key infrastructure or: Assigning roles to strangers," in *Proceedings of the 2000 IEEE Symposium on Security and Privacy*, pp. 2–14, May 2000.
- [15] D.W. Manchala, "Trust metrics, models and protocols for electronic commerce transactions," in *Proceedings of the 18th IEEE International Conference on Distributed Computing Systems*, pp. 312–321, May 1998.
- [16] P. Resnick and R. Zeckhauser, "Trust among strangers in internet transactions: Empirical analysis of ebay's reputation system," in *Proceedings of NBER workshop on empirical studies of electronic commerce*, 2000.
- [17] A. Jsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," in *Decision Support Systems*, 2005.
- [18] P. R. Zimmermann, *The Official PGP User's Guide*, MIT Press, 1995.
- [19] Bin Yu, M. P. Singh, and K. Sycara, "Developing trust in large-scale peer-to-peer systems," in *Proceedings of First IEEE Symposium on Multi-Agent Security and Survivability*, 2004.
- [20] S. Buchegger and J. L. Boudec, "Performance analysis of the confidant protocol," in *Proceedings of ACM Mobihoc*, 2002.
- [21] 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.
- [22] 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.
- [23] D. Gambetta, "Can we trust trust?," in *Gambetta, Diego (ed.) Trust: Making and breaking cooperative relations, electronic edition*, Department of Sociology, University of Oxford, pp. 213–237, 2000.
- [24] A. Jsang and R. Ismail, "The beta reputation system," in *Proceedings of the 15th Bled Electronic Commerce Conference*, June 2002.
- [25] S. Buchegger and J.-Y. Le Boudec, "The effect of rumor spreading in reputation systems in mobile ad-hoc networks," in *Proceedings of Wiop'03*, 2003.
- [26] C. Dellarocas, "Mechanisms for coping with unfair ratings and discriminatory behavior in online reputation reporting systems," in *Proceedings of ICIS*, 2000.
- [27] M. Kinader and S. Pearson, "A privacy-enhanced peer-to-peer reputation systems," in *Proc. 4th International Conference on E-Commerce and Web Technologies*, pp. 206–215, Oct. 2003.
- [28] D. H. McKnight and N. L. Chervany, "The meanings of trust," MISRC Working Paper Series, Technical Report 94-04, Carlson School of Management, University of Minnesota, 1996.
- [29] C. Castelfranchi and Y.-H. Tan, Eds., *Trust and Deception in Virtual Societies*, Kluwer Academic Publishers, 2001.
- [30] T. M. Cover and J. A. Thomas, *Elements of Information Theory*, Wiley-Interscience, 1991.
- [31] J. R. Douceur, "The sybil attack," in *Proceedings of First International Workshop on Peer-to-Peer systems (IPTPS'02)*, 2002.
- [32] J. Newsome, E. Shi, D. Song, and A. Perrig, "The sybil attack in sensor networks: Analysis and defenses," in *Proceedings of the third International Symposium on Information Processing in Sensor Networks (IPSN)*, 2004.
- [33] P. Resnick, R. Zeckhauser, E. Friedman, and K. Kuwabara, "Reputation systems," *Communications of the ACM*, vol. 43, no. 12, pp. 45–48, 2000.
- [34] Michael Kinader, Ernesto Baschny, and Kurt Rothermel, "Towards a generic trust model - comparison of various trust update algorithms," in *iTrust*, pp. 177–192, 2005.
- [35] S. Marti, T. Giulì, K. Lai, and M. Baker, "Mitigating routing misbehavior in mobile ad hoc networks," in *Proceedings of MobiCom 2000*, pp. 255–265, August 2000.
- [36] Y.-C. Hu, A. Perrig, and D. B. Johnson, "Ariadne: A secure on-demand routing protocol for ad hoc networks," in *Proceedings of MobiCom 2002*, Sep 2002.
- [37] W. Liu W. Lou and Y. Fang, "SPREAD: Enhancing data confidentiality in mobile ad hoc networks," in *Proceedings of IEEE INFOCOM'04*, 2004.
- [38] P. Papadimitratos and Z. Haas, "Secure routing for mobile ad hoc networks," in *Proceedings of SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002)*, Jan 2002.
- [39] Wei Yu, Yan Sun, and K.J. Ray Liu, "HADOF: Defense against routing disruptions in mobile ad hoc networks," in *Proceedings of IEEE INFOCOM'05*, March 2005.
- [40] IEEE Computer Society LAN MAN Standards Committee, "Wireless lan medium access control (mac) and physical layer (phy) specifications, IEEE std 802.11-1007," The Institute of Electrical and Electronics Engineers.
- [41] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks, mobile computing," in *Mobile Computing*, edited by Tomasz Imielinski and Hank Korth, Kluwer Academic Publishers, pp. 153–181, 1996.