ISSN 1000-9825, CODEN RUXUEW
Journal of Software, Vol.20, No.9, September 2009, pp.2574−2586
doi: 10.3724/SP.J.1001.2009.00579
© by *Institute of Software, the Chinese Academy of Sciences*. All rights reserved.

E-mail: jos@iscas.ac.cn
http://www.jos.org.cn
Tel/Fax: +86-10-62562563

基于贝叶斯决策的自组网推荐信任度修正模型*

孙玉星[1,2+], 黄松华[1], 陈力军[1], 谢 立[1]

[1](南京大学 计算机软件新技术国家重点实验室,江苏 南京 210093)
[2](南京审计学院 信息科学学院,江苏 南京 211815)

Bayesian Decision-Making Based Recommendation Trust Revision Model in Ad Hoc Networks

SUN Yu-Xing[1,2+], HUANG Song-Hua[1], CHEN Li-Jun[1], XIE Li[1]

[1](State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing 210093, China)
[2](School of Information Science, Nanjing Audit University, Nanjing 211815, China)
+ Corresponding author: E-mail: xyusun2006@gmail.com, http://www.nju.edu.cn

Sun YX, Huang SH, Chen LJ, Xie L. Bayesian decision-making based recommendation trust revision model in ad hoc networks. *Journal of Software*, 2009,20(9):2574−2586. http://www.jos.org.cn/1000-9825/579.htm

Abstract: In this paper, new attacks against trust recommendation are identified and relationships between these attacks are analyzed. Then a trust model is presented with recommendation trust revision support based on Bayesian Decision-Making theory and the minimum- loss principle. The proposed trust model with recommendation trust revision (TMRTR) is employed in ad hoc networks for optimizing ad hoc routing. Simulation results in MATLAB environment show the method proposed is helpful to reduce the impact of some new threats to trust management and improves the effect of trust management. Synchronously, the proposed system can effectively detect malicious nodes in ad hoc networks.

Key words: Bayesian decision-making; trust revision; ad hoc network; trust management; trust model

摘 要: 在分析了信任评估过程中攻击手段及其相互间关系的基础上,提出了基于贝叶斯决策理论的根据推荐偏差度修正对推荐的信任度方法.使用贝塔分布描述推荐偏差度,依据最小损失原则修正对推荐的信任度,并将具备推荐信任修正机制的信任模型运用在自组网的路由协议中,以便优化路由选择.MATLAB 下的仿真结果表明,该方法能够有效抵御一些针对信任管理的威胁并提升信任管理的正确率,进而提高自组网环境下检测恶意节点的效率.

关键词: 贝叶斯决策;信任修正;自组网;信任管理;信任模型
中图法分类号: TP393 **文献标识码**: A

1 Introduction

An ad hoc network is decentralized, self-configured and self-protected. The performance of ad hoc networks depends on the collaboration among distributed entities. These networks provide their entities with higher degree of

* Supported by the National Natural Science Foundation of China under Grant Nos.60573132, 60673154 (国家自然科学基金)
Received 2008-11-21; Accepted 2009-01-20

autonomy and hence are difficult to police. Trust systems have been proposed for a variety of applications, such as making selections of trusted route, detecting misbehaving nodes in ad hoc networks[1].

Many researches[2−4] about trust managements focus on how some local entities can utilize direct evidence and trust recommendation to reach a conclusion about the trustworthiness of some remote entity in a specific situation. Trust management effectively improves the network performance and detects malicious entities, whereas it becomes an attractive target for attackers itself[5−7]. We will firstly show new attacks and relationships between these attacks so as to find the outstanding issues.

By using recommendations, an accurate trust value of some entities' action can be obtained faster. Simultaneously, malicious entities can provide dishonest recommendations to frame up good entities or/and upgrade trust value of malicious entities. This attack, called bad mouthing attack[5], is the most straightforward attack, where attackers are always called liars. This kind of attack can be divided into two categories: one is that the liar gives a bad recommendation when it has a excellent trust value about an entity, the other is that the liar gives a good recommendation when it has a bad trust value about an entity for some especial reasons, such as allying to bad entity. In analyzing the influence of liars, many researches[4−6] focus on one of two categories as the other one is similar by symmetry. In fact, the influences of lying are different in two kinds of bad mouthing attacks, because a mass of false praise is more easily detected than slander. In the extreme cases, if entities around the subject give false laudatory recommendations about the agent, it will find its neighbors' lies through interaction between the subject and agent; if entities around subject give defamatory recommendations about the agent, it will refuse to interact with the agent. As a result, it can't see through neighbors' lies.

In almost all trust models, evaluations of recommendation trust of entity are based on comprehensive considerations of its truth of recommendations for other entities, so this facilitates conflicting behavior attacks[5]. Malicious entities can impair good nodes' recommendation trust by performing differently to different entities. For example, the attacker *K* can always behave well to entity *I* and behave badly to entity *J*. Thus, entity *I* and entity *J* develop conflicting trust values about malicious entity *K*, namely recommendations about entity *K* from entity *J* are conflicting to direct observations about entity *K* gotten by entity *I* itself. As a result, although entity *J* gives true recommendation and should be believed, it will be regarded as a liar in entity *I*'s opinion. There are no effective solutions against the conflicting behavior attack which produces erroneous evaluations of trust recommendations.

The on-off Attack[5] means that malicious entities behave properly for a period of time in order to build up a strongly positive trust, and then begin defecting. This attack exploits the dynamic properties of trust through time-domain inconsistent behaviors. To facilitate the on-off attack on trust management, the observation made long time ago should not carry the same weight as that made recently. The most common way to address this issue is to introduce the weight *u* as a discount factor for past experiences[4,5], which serves as the fading mechanism. Through sun's[5] analysis, the discount factor plays a good role in fighting against the on-off attack. So in this article, there is not much concern about on-off attack.

Through analyses, we know that to enhance security in ad hoc networks, it is important to correctly revise the trustworthiness of recommendations of entities since trust management is the main foundation of collaboration. However, there is few research related to how to revise recommendations trust according to the deviation of recommendation trust for detecting liars. The specific focus of this paper is to provide a mathematical method on how the local entity revises recommendation trust of some entity according to deviations of truth value recommended by some entity form trust value based on observations.

The organization of this paper is as follows. In Section 2 our approach is explained and a mathematical method is proposed for recommendation trust revision according to the deviation of recommendation. In section 3, our trust model with recommendation trust revision (TMRTR) framework is introduced and applied in mobile ad hoc routing protocols. Simulation results are shown in section 4. Section 5 concludes this paper and gives the future work.

2 Solution Proposed

There are many principles for computing trust measures. Arithmetic measures process evidence using simple arithmetic operations; probabilistic measures process trust evidence using some probability-preserving operations[8]. This section addresses measures of trust revision in probability-based trust model in detail.

2.1 Trust model

Although definitions and classifications of trust have been borrowed from social science literature, there is no consensus on the definition of trust in computer networks[9,10]. Trust has been interpreted as reputation, trust opinion, probability, etc.

Definition 1 (**trust**). Trust is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action, both before he can monitor such an action and in a context in which it affects his own action[11].

In our work, trust is established between two parties for a specific action. A trust value is some measure or quantification assigned by one party to its belief in the trustworthiness of the other party. In our work, the first party is referred to as the subject and the second party as the agent, and the trust value is the probability with which the agent will perform an action c in the subject's opinions. Let $\rho_{ij}(c)$ be trust value that is the probability with which subject I believe in agent J with respect to an action c. In other words, $\rho_{ij}(c)=\delta$ represents subject I believe in that agent J will perform an action c with probability δ.

In ad hoc networks, there is no centralized system to manage trust value. Each node maintains its trust records associated with actions. Trust records might be changed in direct or indirect ways[12]. Direct trust is the subject's independent belief in the trustworthiness of agents estimated by direct observations on agents' actions, when direct observations are available. When the subject does not have direct interaction with the agent, it can also establish indirect trust through trust propagation[12]. In our work, we calculate trust propagation using the probability values of trust relationships. The specific methods are similar to Yan's probability-based trust models[5].

2.2 The deviation of recommendation

Definition 2 (**deviation of recommendation**). Let $T_{ik}(c)$ represent the trust value of entity K calculated by entity I through direct observations of target entity K's action c and $R_{ik}^{j}(c)$ represent the same trust value recommended to I by entity J. $D_{ik}^{j}(c)$ represents the deviation $R_{ik}^{j}(c)$ from $T_{ik}(c)$.

$$D_{ik}^{j}(c)=R_{ik}^{j}(c)-T_{ik}(c) \tag{1}$$

$D_{ik}^{j}(c)>0$ means that the recommendation given to I by J is better than the actual one gotten by I itself about K's action c. $D_{ik}^{j}(c)<0$ means that the recommendation is worse than the actual one gotten by I itself about K's action c. The deviation of recommendation is in a range [−1,1].

2.3 Beta distribution of deviation

The beta distribution is used to model the random phenomenon whose value is in a limited range [a,b]. In probability theory and statistics, the beta distribution is a family of continuous probability distributions defined on the interval [a,b] differing in the values of their two non-negative shape parameters, p and q. In this paper, we describe the distribution of random value deviation of recommendation in the beta distribution. The general formula for the probability density function of the beta distribution is

$$f(x)=\frac{(x-a)^{p-1}(b-x)^{q-1}}{B(p,q)(b-a)^{p+q-1}}, a\le x\le b, p,q>0 \tag{2}$$

where p and q are the shape parameters, a and b are the lower and upper bounds, respectively, of the distribution, and $B(p,q)$ is the beta function. The beta function has the formula

$$B(p,q)=\int_0^1 t^{p-1}(1-t)^{q-1}\mathrm{d}t \tag{3}$$

The posteriori (i.e. the updated) distribution of deviation of recommendation is computed by combining the priori (i.e. previous) distribution of deviation of recommendation with the new deviation. As the shape parameters, p and q, directly decide on the shape of the beta distribution from deviation of recommendation, how to get them becomes an important issue. It is assumed that deviations of recommendation have the same credibility, so the sum of p and q is set as a fixed value k.

$$d=\frac{p}{p+q}, k=p+q \tag{4}$$

Thus the parameter p and q are determined as

$$p=dk, q=k-dk \tag{5}$$

where d is deviation of recommendation. We use two beta distributions to depict different liars and to avoid the fact that the alternate lies of two kinds may result in a good recommendation as well.

Definition 3 (**beta distribution of the deviation**). When $D_{ik}^{j}(c)>0$, the random value deviation of recommendation’s probability density functions (PDF) is $f_+(x|p^+,q^+)$; when $D_{ik}^{j}(c)<0$, the random value deviation of recommendation’s PDF is $f_-(x|p^-,q^-)$,where tuple (p,q) comprehends all information about the history of deviations of recommendation. Superscript +, – respectively denote that tuple (p,q) is parameters of f_+ and f_-.

$$f_+(x\mid p^+,q^+)=\frac{x^{p^+-1}(1-x)^{q^+-1}}{B(p^+,q^+)}, 0\le x\le 1, p^+,q^+>0 \tag{6}$$

$$f_-(x\mid p^-,q^-)=\frac{(x+1)^{p^--1}(-x)^{q^--1}}{B(p^-,q^-)}, -1\le x\le 0, p^-,q^->0 \tag{7}$$

Old deviations may not always be relevant to the actual deviation of recommendations, because the entity may change its recommendation over time. What is needed is a model in which old deviations is given less weight than more recent deviations. The forget factor λ can be introduced when all history information of shape parameter (q,p) of deviations is gathered

$$p_{IK(\lambda)}^{J(i)+}=p_{IK(\lambda)}^{J(i-1)+}\lambda+p_{IK(i)}^{J+}, q_{IK(\lambda)}^{J(i)+}=q_{IK(\lambda)}^{J(i-1)+}\lambda+q_{IK(i)}^{J+} \tag{8}$$

where lowercase i indicates how many times the beta distribution is revised, and $0\le\lambda\le 1$.

$$p_{IK(i)}^{J+}=D_{ik}^{j}(c)k; q_{IK(i)}^{J+}=k-D_{ik}^{j}(c)k, D_{ik}^{j}(c)>0 \tag{9}$$

$$p_{IK(i)}^{J-}=k(D_{ik}^{j}(c)+k; q_{IK(i)}^{J-}=-D_{ik}^{j}(c)k, D_{ik}^{j}(c)<0 \tag{10}$$

2.4 Recommendation trust revision

The process of recommendation trust revision is similar to the decision-making process. As stated before, a Bayesian decision model has three elements: (1) a set of situations about the world Θ; (2) a set of decision (action) alternatives A; (3) a preference over the possible outcomes of action: loss function. In our work, we need to revise the recommendation trust according to the posteriori distribution of deviation of recommendation, so three elements are defined as follows:

- Set of Situations Θ: deviation of recommendation θ is in a situation range [–1,1]. If $\theta\ge 0$, d’s PDF is $f_+(\theta|p^+,q^+)$; If $\theta<0$, d’s PDF is $f_-(\theta|p^-,q^-)$.

- Set of Actions *A*: recommendation trust a is in a range [0,1].
- Loss function $L(t,\theta,a)$: calculate the loss of taking recommendation trust a when deviation of recommendation is θ and trust value from observations is *t*.

$$L_{+}(t,\theta,a)=(g(a,(t+\theta))-t)^{2}, \theta\ge 0 \tag{11}$$

$$L_{-}(t,\theta,a)=\theta(g(a,(t-\theta))-t)^{2}, \theta<0 \tag{12}$$

where $g(\cdot)$ represents the true propagation function for inference and *t* is known. L_{+} represents Loss function for false praise and L_{-} represents Loss function for slander.

Having set of situations, set of actions and the loss function, decision-making means to decide a recommendation trust which produces the minimum expectation loss. We will get a_{+} and a_{-} respectively from the process of seeking the minimum expectation loss function $\varphi_{+}(a)$ and $\varphi_{-}(a)$.

$$\varphi_{+}(a)=\int L_{+}(t,\theta,a)f_{+}(\theta\mid p^{+},q^{+})\mathrm{d}\theta, \theta\ge 0 \tag{13}$$

$$\varphi_{-}(a)=\int L_{-}(t,\theta,a)f_{-}(\theta\mid p^{-},q^{-})\mathrm{d}\theta, \theta<0 \tag{14}$$

Finally, recommendation trust will be decided as

$$\begin{cases} a_{+}, & \dfrac{p^{+}+q^{+}}{p^{-}+q^{-}}\ge r \\ a_{-}, & \dfrac{p^{-}+q^{-}}{p^{+}+q^{+}}\ge r \\ \dfrac{(p^{-}+q^{-})a_{-}+(p^{+}+q^{+})a_{+}}{p^{-}+q^{-}+p^{+}+q^{+}}, & \text{else} \end{cases} \tag{15}$$

To decide whether to synthesize two recommendation trusts, *r* is the threshold which reflects the difference in the amount information of two beta distributions.

3 Applications in Route Optimization

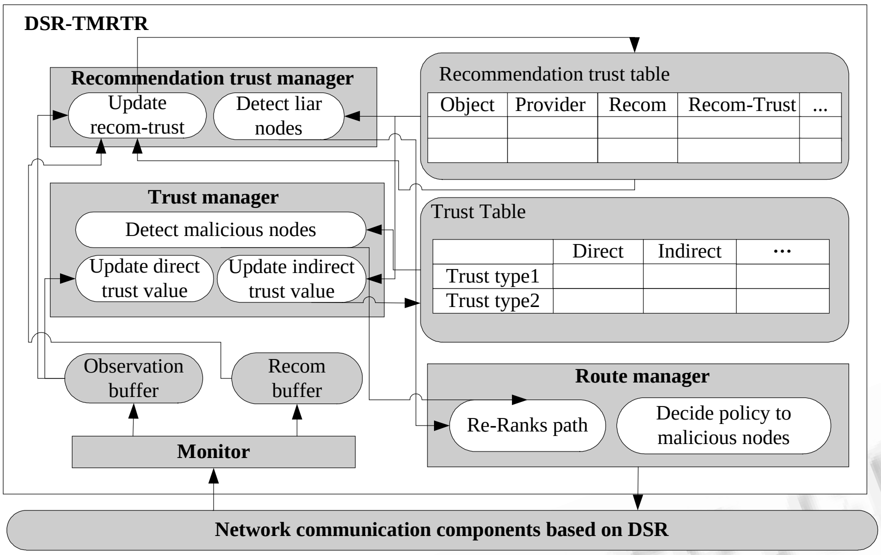
In ad hoc networks, most secure routing protocols, such as Ariadne[13], SEAD[14], SRP[15] focus on preventing attackers from modifying routing data to ensure the accuracy of routing information. However those protocols have no mechanism to detect packets dropped by selfish/malicious nodes. Trust management can guard routing even if malicious nodes have gained access to the network.

Many trust/reputation managements are studied to mitigate the detrimental effect of selfish/malicious nodes. Some of them[16,17] use direct observation only and disallow disseminations of second-hand trust information. Although they eliminate most trust management complexity, the accuracy of trust values becomes lower and they need some message to alarm neighbors who are malicious nodes. Malicious nodes may broadcast the false alarm messages to hinder correct routing. Others[2−5] use second-hand trust information (recommendation information), but they do not find good solutions to the attacks using recommendation information.

Our system is based on Dynamic Source Routing Protocol (DSR)[18], using TMRTR to find out and avoid malicious nodes. TMRTR is used to evaluate the forwarding behavior according to the network protocol. Routing decisions are based on DSR route protocols and trust value of forwarding behavior of other nodes.

3.1 Design of DSR-TMRTR

With DSR-TMRTR, each node has the following four components: a monitor, a trust manager, a recommendation trust manager and a route manager. These components interact with each other to provide and process protocol information, as shown in Fig.1.


DSR-TMRTR
Recommendation trust manager
Update recom-trust
Detect liar nodes
Recommendation trust table
Trust manager
Detect malicious nodes
Update direct trust value
Update indirect trust value
Trust Table
Observation buffer
Recom buffer
Route manager
Re-Ranks path
Decide policy to malicious nodes
Monitor
Network communication components based on DSR

Object	Provider	Recom	Recom-Trust	...

	Direct	Indirect	···
Trust type1			
Trust type2			

Fig.1 Components of DSR-TMRTR

The monitor is the equivalent of a “neighbor watch”, where nodes locally monitor neighbors’ forwarding behavior. The monitor reports all events including forwarding packets or dropping packets to the trust manager. Besides monitoring forwarding actions between the neighbors, the monitor also reports these nodes’ reaction to its own requests.

The trust manager establishes and updates the trust value of forwarding packets according to direct observations or indirect trust values in the recommendations. Outdated trust values will be cleared by the trust manager. Trust manager detects malicious nodes based on trust value and reports these malicious nodes to the route manager.

The route manager re-ranks path according to the trust value of the nodes in the path and decides how to respond, like ignoring requests from malicious nodes.

The recommendation trust manager requests recommendations from other nodes and updates the recommendation trust values of other nodes according to the deviation of recommendation provided by them.

3.2 Process of routing with TMRTR

For ad hoc routing, we deal with the trust values related to two actions: Forwarding packets and making recommendations. In other words, each node maintains trust values of other nodes associated with two actions. When a source wants to establish a route to the destination, our system finds multiple routes with changed DSR route protocols. Instead of replying to the first route request, the destination node replies to all route requests it received by sending a route reply with the accumulated route. When the source node receives multiple routes, the source will measure packet-forwarding trustworthiness of nodes on different paths from its own trust values and recommendations of other nodes. Lastly, the source selects the trustworthy route to transmit data.

3.2.1 Trust value maintenance and update

In this system, the trust value of forwarding packets $T_{sa}(forward)$ is established based on whether node A forwards packets to node S and whether node A forwards packets to other nodes if A is the neighbor of node S. Assume that node S asks node A to forward n packets and node A actually forwards m packets, then trust value $T_{sa}(forward)=(m+1)/(n+2)$. Simultaneously, node S observes forwarding actions between neighbors of node S. Every time node I forwards a packet, the neighbor of node I who observes this forwarding increases numerator and denominator of T_{ki} by 1, in which $k \in N(i)$ (the set of neighbors of node I) who observes this forwarding action. Similarly, if the neighbor of node I who observes that node I drops a packet, then denominator of T_{ki} will be increased by 1. Briefly speaking, the trust value of forwarding of a node I is updated according to not only the number of packets node I forwarded to estimator but also the number of packets node I forwarded to other nodes which could be monitored by estimator. In this way, the estimator may evaluate trust values of other nodes more objectively.

3.2.2 Recommendation trust revision

In DSR-TMRTR, each node saves the trust value of forwarding packet action and that of recommendation provided by other nodes. To conflict against behavior attacks, the truth of recommendation is calculated based on not only the provider of these recommendations but also the object of these recommendations. So the object and provider of recommendations are saved in recommendation trust table. Every REVISION_TIME, recommendation trusts are revised according to direct observations information in this REVISION_TIME. More details of process of recommendation trust revision are showed in Section 2.

3.2.3 Obtaining trust recommendation

A node disseminates a trust recommendation request (TRRQ) message when it needs a trust value about other nodes and does not have one available. This will happen when it hasn’t any direct observation information about that node or the direct observation information becomes invalid.

After broadcasting a TRRQ, a node waits for trust recommendation reply (TRRP) messages. If a TRRP is not received within NET_TRAVERSAL_TIME milliseconds, the node may try again to discover trust recommendation by broadcasting another TRRQ, up to a maximum of TRRQ_RETRIES times at the maximum time-to-live (TTL) value. Each new attempt must increment and update the TRRQ ID. To prevent unnecessary network-wild dissemination of TRRQs, the originating node should initially use a *TTL=TTL_START* in the TRRQ packet IP header. If TRRQ times out without a corresponding TRRP, the originator broadcasts the (TRRQ) with TTL incremented with 2. In this way, the TTL field of the IP header is used to control how far the TRRQ is disseminated for each retry.

When a node X receives a TRRQ, it will send trust value to the originator Y if it has trust value of forwarding of the node B. Otherwise, X will forward this TRRQ message to X’s neighbors by reducing the value of TTL by one. Node X uses corresponding recommendation trust values to establish trust propagation paths for calculate B’s trust values of forwarding and return the result to Y. More details of this mechanism are presented in Ref.[5]. By the way, trust value information is saved in B.

4 Solution Evaluation

4.1 Parameters determination

In this section, to determine the parameters k and λ, we study their impacts on the value of recommendation trust by simulations. Figure 2 represents the estimations of value of recommendation trust as a function of deviation of recommendation for k being 20, 30, 60, 80 and trust value being 0.5. It can be seen that the greater k is, the sooner the value of recommendation trust declines with the increase in the degree of deviation. The recommendation trust when k=60 is close to that when k=80, so k has smaller impact on the value of recommendation trust when k>=60.

In Fig.3, we show the value of recommendation for k being 20, 30, 60, 80 and trust value being 0.5 and λ=0.6, when the deviation of recommendation is increasing by the addition of deviation is 0.025 a time. From Fig.3, parameter *k* has some small impact on the value of recommendation trust but at stages of start and end of the change of deviation, simultaneously the calculation results when *k*=60 are very close to the calculation results when *k*=80. So in the other simulations, the parameter *k* is set 60.

Next, we discuss the dynamic properties of the relationship between the value of recommendation trust and parameter λ. Assume a liar behaves in the following three stages: (1) provide recommendations without any deviation for 50 times; (2) provide recommendations with 0.25 deviation for 50 times; (3) provide recommendations with 0.5 deviation for 50 times.

From Fig.4, we observe that:

1. When the system does not forget the history information, i.e. λ=1, the liar has positive recommendation trust, and the value of which is closer to 1 with the increase in times in stage (1). When using a large forgetting factor, the recommendation trust value may not represent the latest status of liar, and the recommendation trust can't lower with the increase of deviation in stage (2) and stage (3). As a consequence, the liar could cause a large amount of damage in the next two stages.
2. When using a small forgetting factor, the recommendation trust of liar drops rapidly after its deviation becomes bigger in stage (2) and stage (3). However, continued recommendation without deviation can't improve recommendation trust in stage (1). Similarly, it can regain recommendation trust by providing a true recommendation for just one time.

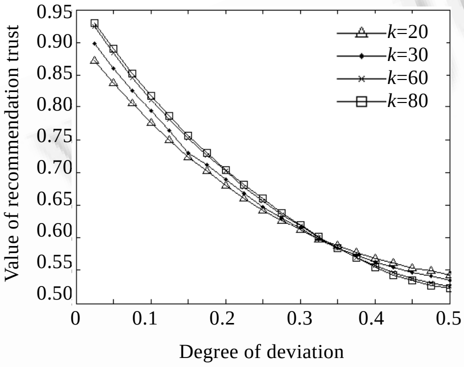


Fig.2 Value of recommendation trust vs. parameter *k* in the static changes of deviation

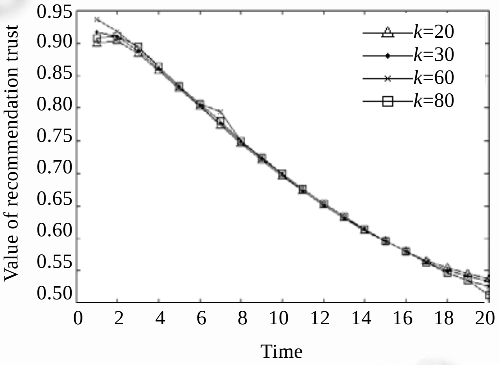


Fig.3 Value of recomendation trust vs. parameter *k* in the dynamic changes of deviation

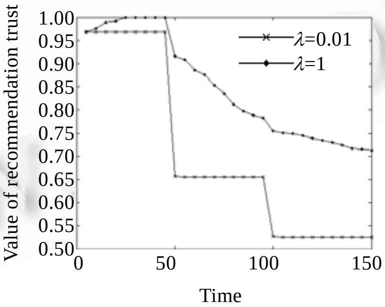


Fig.4 Value of recommendation vs. parameter λ

Instead of using a fixed forgetting factor, λ is a function of the current recommendation trust value. For example, we can choose λ=1−*a*, where *a* is the recommendation trust value. In this way, recommendation trust value can keep up with the entity's current status after the entity provids false recommendations. And an entity can recover its recommendation trust value after some false recommendations, but this recovery requires many true recommendations[5].

4.2 Performance analysis

To compare with other trust management without using recommendation, a simulation was implemented using NS2 to evaluate the performance of DSR-TMRTR. The MAC layer protocol simulates the IEEE 802.11 Distributed Coordination Function. DSR[18] is used as the routing algorithm. The Dimension of space is size 1000m by 1000m and the network size is about 100 nodes. The maximum radio range is 250m. Each node moves randomly according to the random waypoint model[18], in which nodes move to a random destination and once they reach the destination, short pauses are introduced between changes in direction or speed.

For the performance analysis, we compare the behavior of DSR-TMRTR to other trust system without using recommendation. In this analysis, a metric packet delivery ratio is used to describe the performance of routing protocol with trust management. The ratio of the data packets delivered reflects the degree of successful ad hoc network communications. Note that attacks mentioned in this paper are not considered because trust system without using recommendation hasn't similar problems. Packet forwarding ratio[16] is defined by

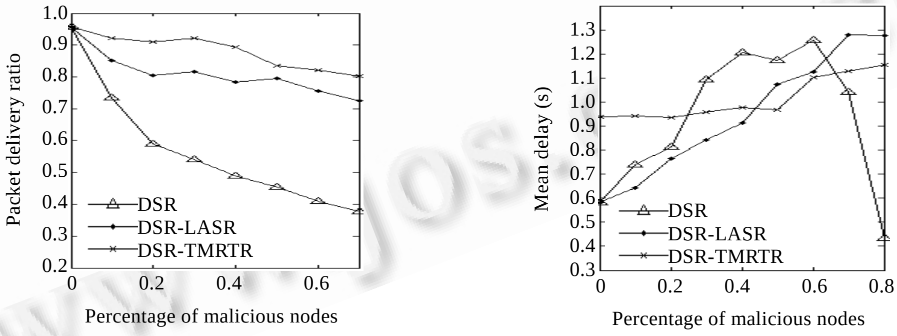
$$P_{delivery}=\frac{\sum packet_{recieved}}{\sum packet_{sent}} \tag{16}$$

A metric mean delay of data packets is also used to describe the efficiency of ad hoc network transmission. It is defined as

$$D=\frac{\sum d}{\sum packet_{recieved}} \tag{17}$$

The parameter d indicates how long a packet forwards to destination.

Figure 5(a) shows the mean delivery ratio for varying percentage of malicious nodes in the ad hoc network. The delivery ratio in DSR-TMRTR sustains about 80% of the original value while it falls dramatically in the network without TMRTR. This is because TMRTR helps the network to identify and isolate the malicious nodes. Compared with TMRTR, LARS has a lower packet delivery ratio because it only considers direct observation and disallows exchanging recommendations, in this way, the accuracy of detecting malicious nodes can be lower than TMRTR.



(a) (b)

Fig.5 Delivery and mean delay for varying percentage of malicious nodes

Figure 5(b) shows the mean delay for varying percentage of malicious nodes in the ad hoc network. With the varying percentage of malicious nodes, the mean delay in DSR-TMRTR is more stable than LARS and DSR. Due to the regular recommendation exchange and recommendation trust revision, the mean delay in TMRTR is larger than

LARS. However, due to the slower detection speed, the delay in LARS is larger than that in TMRTR when the percentage of malicious nodes is great.

4.3 Attack-Resistance analysis

We evaluate the performance of our recommendation trust revision method by simulation in this section. The motivation of trust revision is to ensure the quality of trust management, even in face of new attacks previously mentioned. All simulations were implemented in Matlab 7 (Mathworks Inc, Natick, USA). To investigate the effect of TMRTR on the resistance to the bad mouthing attack and conflicting behavior attack, simulations were run on small world network topology: the total number of nodes is 100, a few of which have a high degree, and all the rest have fewer neighbors. The average degree is 8, but the highest is 20. The small world topology for trust has been used in building reputation for MANETs in Ref.[19]. To create the (non-hierarchical) small-world networks, we used a procedure similar to the algorithm presented in Ref.[20].

4.3.1 Against bad mouthing attack

First, we study a network in the presence of bad mouthing attacks, i.e. there are untrustworthy nodes who give true or false recommendation about other nodes. Untrustworthy nodes may have different ways to publish their falsified recommendations to affect the production of trust value, e.g. when they want to discredit normal nodes or raise the trust value of misbehaving nodes. Similarly, there are many different relationships between untrustworthy entities. Three types of relationships are considered: 1) Independent: liars do not collude with each other. They provide good recommendations to all entities independently; 2) Collusive: liars know each other. They always give their friends good recommendations and give other entities bad recommendations; 3) Random: liars randomly give other entities good or bad recommendations at each time.

For easy presentation, we explicate precisely the liar strategies in a mathematical way by providing metrics to depict degree of lies. Table1 describes the general simulation setup, including the node types, behavior strategies.

We introduce a metric Correct Probability $P_{correct}$ to describe the performance of trust management. Let $\rho_{ij}(c)$ be the probability with which node *I* believes in node *J* with respect to an action *c*. Let $\mu_j(c)$ be the probability with which node *J* performs an actual action *c*. *N* denotes the set of all nodes. M_i denotes the set of nodes of which trust values have been calculated in node *I*. $P_{correct}^{i}$ denotes node *I*'s Correct Probability.

$$P_{correct}^{i}=\frac{\sum_{j\in M_i}\left(1-\frac{|\rho_{ij}(c)-\mu_j(c)|}{\mu_j(c)}\right)}{|M_i|} \tag{18}$$

$$P_{correct}=\frac{\sum_{i\in N}P_{correct}^{i}}{|N|} \tag{19}$$

Table 1 Parameters and behavior strategies in simulation

	Parameter	Value
Node model	Number of nodes in the network	100
	Percentage of good nodes	10%~90%
	Percentage of untrustworthy nodes	0%~80%
	Percentage of malicious nodes	0%~10%
	Deviation of lies	0.5, 0.9
Behavior strategies	Good nodes: Always provide truthful recommendation about others nodes Untrustworthy nodes: • Independent: Provide good recommendation to all nodes • Collusive: Provide bad recommendation to good nodes and provide good recommendation for untrustworthy nodes and malicious nodes • Random: Provide false recommendation to random nodes Malicious nodes: Give bad services and provide bad recommendation for good nodes and provide good recommendation for untrustworthy and malicious nodes	

Figure 6 shows the Correct Probability for six cases when the percentage of untrustworthy nodes changes from 0% to 80%. From Fig.6, we can see that when the deviation is 0.9 and behavior strategy is collusive or random, the correct probability sharply drops because some nodes will give bad recommendation to other nodes. With the increasing percentage of untrustworthy nodes, an increasing number of nodes will get error value trust from trust management due to the bad recommendations provided by all neighbors and deviation of recommendation preventing to find out these liars.

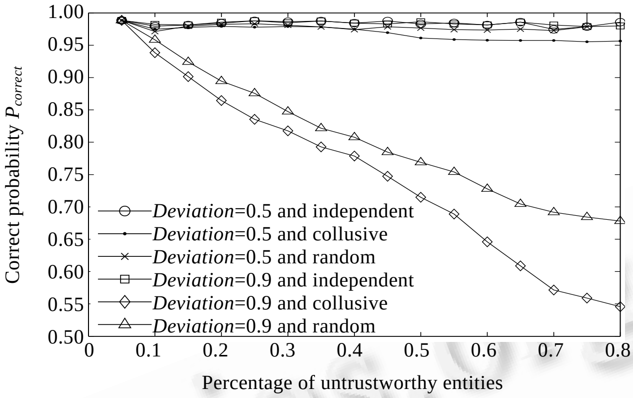


Fig.6 $P_{correct}$ vs. percentage of untrustworthy nodes

Figure 7 shows the process of Correct Probability reaching steady states for seven cases when the percentage of untrustworthy nodes is fixed 40%. In case 7, the deviation of lies is 0.5 and behavior strategy is independent without our trust revision system. From Fig.7, we know trust management with recommendation trust revision will quickly reach stable states compared with the one without recommendation trust revision. Synchronously, the greater the deviation of recommendation is, the shorter the time is required to reach stable states.

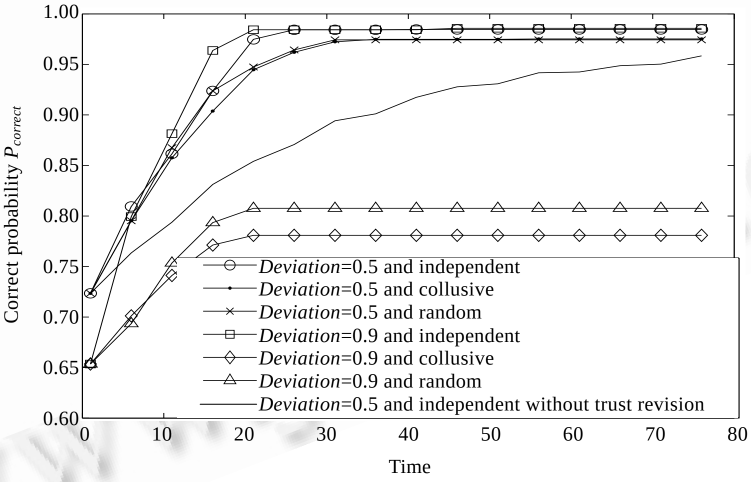


Fig.7 $P_{correct}$ vs. time

4.3.2 Against conflicting behavior attack

Now, we study a network in the presence of conflicting with behavior attacks. In this simulation, malicious nodes impair good nodes' recommendation trust by performing differently to different nodes. We introduce a metric Degree of Impact on Trust evaluation to describe the impact of conflicting behavior attacks on trust management.

Degree of Impact on Trust evaluation (DIT) synthetically considers the changes of efficiency of recommendations provided by the attacked entities. We compare the performance of a trust management with recommendation trust revision and one without revision when the deviation of recommendation is 0 and malicious entities provide conflict servers in Fig.8. It can be seen that the trust management with recommendations trust revision reduces the impact of malicious on the trust evaluation further.

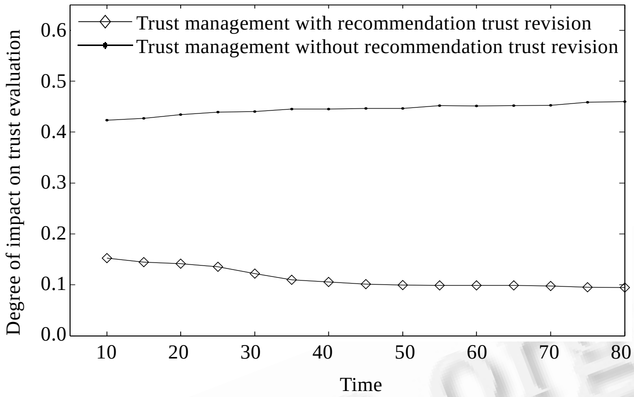


Fig.8 DIT vs. time in conflicting behavior attacks

In our work, instead of comprehensively considering the truth of recommendations for other nodes provided by the one whose recommendation trust is assessed, the truth of recommendation is calculated based on not only the provider of these recommendations but also the target of these recommendations. In other words, instead of calculating recommendation trust $\rho_{ij}(r^c)$, we evaluate recommendation trust of object *K* $\rho_{ij}(r_k^c)$ provided by entity *J*, subscript *k* representing which object is recommended. By distinguishing targets recommended in evaluations of recommendation trust of entity *J*, we observe whether recommendations provided by entity *J* for most objects or for just one object are conflict with entity *I*’s direct observations. In this way, the conflicting behavior attacks by malicious entities will have smaller impact on the trust management than other counterparts.

5 Conclusions

To deal with new attacks on trust management for ad hoc networks, we need to reduce the impact of liars by revising their recommendation trustworthiness. In this paper, we proposed a trust management model with recommendation trust revision for ad hoc networks. Our solution, which is based on the theory of Bayesian decision, uses the beta distribution to depict the deviation of recommendation, and revises the recommendation trust according to the distribution. The approach is fully distributed and no agreement is needed, which is suitable for ad hoc networks. The proposed trust model with recommendation trust revision (TMRTR) is employed in ad hoc networks for securing ad hoc routing. The simulation results show that the routing protocols with TMRTR can improve network throughput compared with the trust management without using recommendation for its exact and quick detection of malicious nodes. Our simulation also shows the impact of lies on trust evaluation is reduced quickly, and our model resists the new attacks on trust management. But as we can see, if liars are in random behavior strategy or new attacks provide mixed impact on trust management, the performance of trust management will drop greatly. So for future work, we plan to modify the trust revision model to adapt to the need of withstanding the joint effects of new attacks.

References:

[1] Li XQ, Lyu MR, Liu JH. A trust model based routing protocol for secure ad hoc networks. In: Proc. of the Aerospace Conf. IEEE Computer Society Press, 2004. 1286−1295.

[2] Theodorakpoulos G, Baras JS. On trust models and trust evaluation metrics for ad-hoc networks. IEEE Journal on Selected Areas in Communications, 2006,24(2):318−328.

[3] Sun Y, Yu W, Han Z. Information theoretic framework of trust modeling and evaluation for ad hoc networks. IEEE Journal on Selected Areas in Communications, 2006,249(2):674−679.

[4] S.Buchegger, Le Boudec JY. Self-Policing mobile ad hoc networks by reputation systems. IEEE Communications Magazine, 2005, 43(7):101−107.

[5] Sun YL, Han Z, Liu KJR, Defense trust management vulnerabilities in distributed networks. IEEE Communications Magazine, Feature Topic on Security in Mobile Ad Hoc and Sensor Networks, 2008,46(2):112−119.

[6] Sun YL, Han Z, Yu W, Liu KJR. Attacks on trust evaluation in distributed networks. In: Proc. of the Information Sciences and Systems. 2006. 1461−1466.

[7] Azer MA, El-Kassas SM, Hassan AWF, El-Soudani MS. A survey on trust and reputation schemes in ad hoc networks. In: Proc. of the 2008 3rd Int'l Conf. on Availability, Reliability and Security. IEEE Computer Society Press, 2008. 881−886.

[8] Glenn R. Mahoney: A generalized trust model using network reliability [MS. Thesis]. University of Victoria, 2004.

[9] Jøsang A. Ismail: A survey of trust and reputation system for online service provision. Preprint of article published in Decision Support Systems, 2007,43(2):618−644.

[10] Jøsang A. Trust and reputation systems. In: Aldini A, Gorrieri R, eds. Proc. of the Foundations of Security Analysis and Design IV, FOSAD 2006/2007 Tutorial Lectures. Springer-Verlag, 2007. 209−245.

[11] Diego G. Can we trust trust? In: Diego G, ed. Proc. of the Trust: Making and Breaking Cooperative Relations. Blackwell, 1990. 213−237.

[12] Feng YF. Adaptive trust management in MANET. In: Proc. of the 2007 Int'l Conf. on Computational Intelligence and Security. 2007. 804−808.

[13] Hu YC, Perrig A, Johnson D. Ariadne: A secure on-demand routing protocol for ad hoc networks. Wireless Networks, 2005,11(1-2): 21−38.

[14] Hu YC, Johnson D, Perrig A. SEAD: Secure efficient distance vector routing for mobile wireless ad hoc networks. In: Proc. of the IEEE Workshop on Mobile Computing Systems and Applications (WMCSA). IEEE Computer Society, 2002. 3−13.

[15] Papadimitratos P, Haas Z. Secure link state routing for mobile ad hoc networks. In: Proc. of the IEEE Workshop on Security and Assurance in Ad hoc Networks. IEEE Computer Society, 2003. 379−384.

[16] Hu JY. Trust management in mobile wireless networks: Security and survivability [Ph.D. Thesis]. Florida State University, 2007.

[17] Bansal S, Baker M. Observation-Based cooperation enforcement in ad hoc networks. Technical Report, Arxiv Preprint cs. NI/0307012, Stanford University, 2003.

[18] Johnson DB, Maltz DA. Dynamic source routing in ad hoc wireless networks. In: Imielinski T, Korth H, eds. Proc. of the Mobile Computing. Boston: Kluwer Academic Publishers, 1996. 153−181.

[19] Sankhla V. SMART: A small world based reputation system for MANETs [MS. Thesis]. University of Southern California, 2004.

[20] Watts DJ, Strogatz SH. Collective dynamics of 'small-world' networks. Nature, 1998,393(4):440−442.

SUN Yu-Xing was born in 1977. She is a Ph.D. candidate in the Department of Computer Science & Technology at Nanjing University. Her research areas are wireless networks and network security.

CHEN Li-Jun was born in 1961. He is a professor at Nanjing University. His research areas are distributed computing and wireless sensor networks.

HUANG Song-Hua was born in 1979. He is a Ph.D. candidate at the Nanjing University and a CCF student member. His current research areas are network mobility and network security.

XIE Li was born in 1942. He is a professor and Ph.D. supervisor at Nanjing University and a CCF senior member. His current research areas cover information security, the parallel and distributed computing.