Rendezvous Based Trust Propagation to Enhance Distributed Network Security

Ningning Cheng*

Department of Computer Science,
University of California, Davis,
CA 95616, USA
E-mail: chengni@cs.ucdavis.edu
*Corresponding author

Kannan Govindan

Department of Computer Science,
University of California, Davis
CA 95616, USA
E-mail: gkannan@cs.ucdavis.edu

Prasant Mohapatra

Department of Computer Science,
University of California, Davis,
CA 95616, USA
Fax: (530) 752-4767
E-mail: prasant@cs.ucdavis.edu

Abstract: Development of network of nodes connected with their trust values and the propagation of these trust values to far away nodes are basic operations of the modern day trustworthy networks. Trust can be exploited to mitigate the security threats in wireless network. Most of the existing trust propagation methods are based on flooding trust information, which puts a heavy burden on wireless communication, especially in ad hoc network and sensor network. In this paper, we propose a rendezvous based trust propagation scheme. Trust requester and trust provider send out trust-request and computed-trust tickets respectively, which will meet in some common rendezvous node with certain probability. Computed-trust will then be propagated to the requester. We carry out detailed performance evaluations of our scheme. The results show that our method achieves up to 66% overhead reduction in trust propagation compared to flood based methods.

Keywords: trust propagation; ad hoc network; rendezvous; network security.

Reference to this paper should be made as follows: Cheng, N., Govindan, K. and Mohapatra, P. (2011) ‘Rendezvous Based Trust Propagation to Enhance Distributed Network Security’, *special issue of International Journal of Security and Networks*, Vol. 1, Nos. 3/4, pp.197–212.

Biographical notes: N. Cheng received her B.Sc, M.Sc degree in the department of Computer Science and Technology in Nanjing University, China. Currently she is doing her PhD in Computer Science at University of California, Davis. Her areas of interest are computer networks and distributed systems, with a focus on network management, and security.

K. Govindan received his BE degree from NIT-Trichy India in 2002 and PhD from IIT-Bombay India in 2009. He worked as scientist for Indian defence R and D from 2002-2004 and as Researcher for General Motors Research India from 2008-2009. Since 2009-May he have been working as postdoc in Networks lab of University of California, Davis. His research interest includes network systems and network security.

Dr. Prasant Mohapatrpras is currently a Professor in the Department of Computer Science at the University of California, Davis. Dr.Mohapatra received his Ph.D. in Computer Engineering from the Pennsylvania State University in 1993. He was/is on the editorial board of several IEEE and ACM sponsored journals and conferences. Dr. Mohapatra’s research interests are in the areas of wireless networks, sensor networks, Internet protocols and QoS.

1 Introduction

When a set of distributed entities collaboratively participate in a certain activity, the concept of trust can be abstracted from their relationships to predict future behaviors in the activity. Trust effectively helps to improve the security in the network [S. 06]. If a node gets the trust information of other nodes in advance, it can avoid communicating with untrustworthy neighbors or cooperating with dishonest partners and hence reduce the chance for misbehaviors. This way we can have a set of trustworthy nodes in the network and ensure successful network operations.

In a distributed network, such as wireless ad hoc network or sensor network, trust computation inherently requires distributed calculation. Either the result of trust computation needs to be propagated from the provider to the requester, or the trust query needs to be transported from the requester to the provider in a distributed way. In the existing works, a widely accepted method of trust propagation is flooding. Trust requesters send out recommendation requests when trust information is needed. After receiving the request, the set of nodes which can provide trust information will transmit it to the requesters. Then, the recommendation path will be set up from one of the recommenders to the requester. In the end, the final trust value will be aggregated from different recommendation paths. The recommendation path constructing phase is very important. An efficient algorithm will help provide fast bootstrapping. Although flooding is simple and easy to deploy, the major concern is the flooding overhead [C. 05, Y. 05]. It increases exponentially by path length. When a trust information provider is far away from the request node, the communication overhead is very heavy. To the best of our knowledge, recent studies focus only on minimizing the recommendation path, where requester takes as few hops as possible to get the trust recommendation. However, it is possible that the requester is far away from any provider node and no short path exists. In addition, due to distributed property, trust requesters seldom have knowledge on trust providers. Therefore, provider discovering is needed before trust information is being delivered to the requester, which prolongs the propagation delay.

In this paper, we propose a rendezvous based trust propagation scheme to solve issues associated with trust propagation. The communication overhead cost is reduced. Instead of notifying trust information by the provider, the notification of trust information can also be issued by a third party node, the rendezvous node. There are three parties in our approach:

- ***Target***, the node whose trust information is inquired in some applications.

- ***Requester***, the node who inquires the trust information of *target*.

- ***Provider***, the node who can provide the trust information of *target* to *requester*.

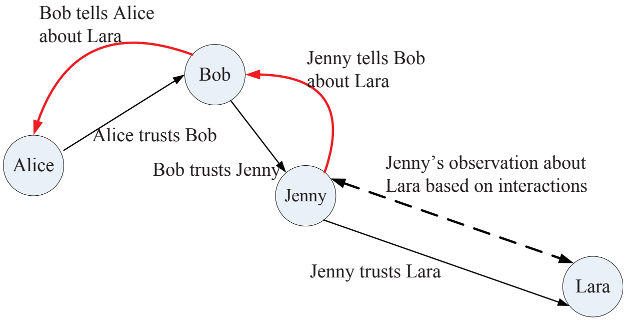

Bob tells Alice about Lara
Bob
Jenny tells Bob about Lara
Alice
Alice trusts Bob
Bob trusts Jenny
Jenny
Jenny's observation about Lara based on interactions
Jenny trusts Lara
Lara

Figure 1 Propagation of trust in a simple straight chain

For instance in Fig. 1 suppose *Alice* received a message from far away node *Lara*. To evaluate the trustworthiness of this message, requester *Alice* wants to know the trustworthiness of target *Lara*. However *Lara* is out of its communication range and no direct trust information is available; *Jenny* who is neighbor of *Lara* has direct trust relationship with *Lara* and knows *Lara* is a trustworthy node. But being the trust provider, *Jenny* does not know who need the trust information of *Lare*; *Bob*, who can communicate with both *Alice* and *Jenny* knows that *Alice* is search for a trust provider and *Jenny* is search for a trust requester. Then, provider *Jenny* can communicate with *Alice* through a one hop neighbor *Bob*. In this case, *Bob* becomes the rendezvous node since it knows both the requester and the provider.

The virtual of rendezvous-based trust propagation lies in its simplification. It does not need network topology information to find out trust providers or requesters. Also it reduces the communication overhead without compromising too much accuracy in trust prediction.

We make the following assumptions in our paper:

- Each node is able to monitor its neighboring nodes' cooperation behaviors.
- Nodes' behaviors are consistent. The good nodes will always report honestly and behave cooperatively. The malicious nodes can report dishonest message and be uncooperative and selfish.
- The majority of the network member is good. There are no consecutive malicious nodes along a single communication path.
- There is no collusion attacks in the network, which means all the malicious nodes are working independently for their own interests and do not share information with each other.

This paper is organized in the following way: Section 2 reviews some of the related work in the trust propagations. Foundation for the proposed trust propagation scheme is provided in Section 3. Our proposed trust propagation protocol is presented in detail in Section 4. Performance evaluation of the proposed scheme is carried out in Section 5. Concluding remarks are given in Section 6.

2 Related work

Trust propagation in small world network is proposed in [H. 04]. In small world network when the nodes form a trust propagation path, it is relatively short due to the small world influence. This approach can be used only in certain specified self-organized ad hoc network where network diameter is very short.

Trust propagation in social networks have been studied in [G-F10, C-W09, U. 07, D. 07, R. 04, J. 05]. These work focus more on trust concatenation, aggregation and path selection in a social semantic web, which differs from wireless communication, where transmission overhead and propagation delay are the major issues.

Traditional trust propagation in a distributed wireless ad hoc network is usually based on different recommendation paths [F. 08, ABC$^+$08, OS05]. Since the entity does not know who has evaluated the objects' trustworthiness, recommendation request is distributed along direct neighbors, which is basically flooding trust requests.

Propagation of the security credentials such as cryptography keys, trust information by exploiting mobility is analyzed in [S. 03], where nodes exchange trust information as soon as they are connected. Performance of this strategy depends on the mobility pattern, density of the nodes and other related parameters. Trust propagation based on spreading activation models is proposed in [ZL04, ZL05]. Spreading activation is a method for searching trust values or any intended values of nodes in the networks.

Rendezvous-based method has been studied in previous literatures such as information retrieval in p2p networks [W. 07b] or geographic routing in sensor networks [S. 05]. These previous works shows that rendezvous-based method is well suited to content specific and structure-free networks. The difference between these work and our study is that we not only uses rendezvous node as a refer node for the data source, but also considers trust calculation and aggregation along the recommendation path from the rendezvous to the requester.

Our approach considerably differs from the above stated trust propagation work as we do not use flooding based techniques to avoid overhead. A preliminary version of our work is introduced in [N. 11]. In this paper, we improve our approach both in theoretical details and experimental studies. We also extend the fixed-ticket trust propagation to an adaptive trust propagation which can further reduce the ticket number in use.

3 Foundation of the propagation model

3.1 Terminology

In our paper, trust and trustworthiness have similar meanings. Trust reflect one party (trustor)'s willingness to be dependant on another party (trustee). Trustworthiness is the estimation of one party (trustee)'s worthiness in the eye of another party (trustor). Therefore, we use trust and trustworthiness interchangeably. In order to give a formal description of trust and trustworthiness, we give our definitions of trust/trustworthiness as follows:

Definition 3.1: The trustworthiness(Trust) of node n: The trustworthiness of a node n, denoted as $T(n)$, is the probability that n behaves consistently over the time and forwards/generates correct information. In this paper, we use the term "trust" to represent the trustworthiness of a node.

Definition 3.2: The trustworthiness of node B evaluated by node A: The trustworthiness of node B evaluated by node A, denoted as $T(A, B)$, is the probability that B behaves consistently over the time and forwards/generates correct information during A's observation. It reflects the trustworthiness of B in the eye of beholder A.

Trustworthiness of some information is not only decided by the trust value provider, but also decided by how this information is propagated. Based on this observation, we define the trustworthiness of a communication path as follows:

Definition 3.3: The trustworthiness of a communication path P_{AB}: The trustworthiness of a path, denoted as $T(P_{AB})$, is the accumulative trustworthiness of every trust information forwarder along the path. Let us assume B is the trust information requester, A is the trust information provider and node A transmits the trust value through path P_{AB}. The trustworthiness of a communication path is the probability that the trust information calculated along path P_{AB} reflect the correct trust information in B's observation.

In trust computing system, the recommendation path usually serves as the communication path, too. Therefore, we use communication path and recommendation path interchangeably in the rest of our paper. To spread trust messages we use tickets. Tickets are small packages containing observers' ID, the ID of the node being observed and a time stamp. We define the following two types of tickets to propagate trust message:

TR ticket: Trust Request ticket.

CT ticket : Computed Trust ticket.

Trust requester disseminate TR tickets indicating they are interested in some node's trust value. Nodes receiving TR tickets can reach trust requesters based on TR ticket routing path; On the other hand, trust provider disseminate CT tickets representing the trust information it can provide. Nodes receiving CT tickets can reach trust providers based on CT ticket routing path. The node that receives both the CT ticket and

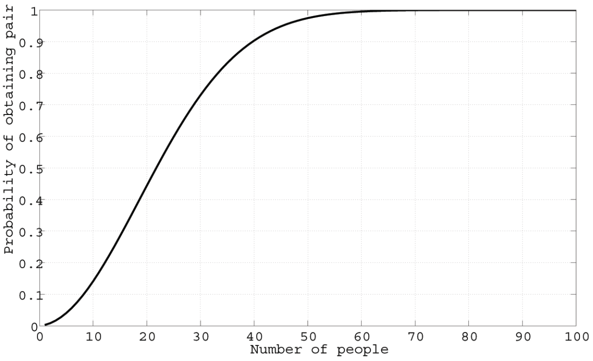


Figure 2 Probability of at least two people getting common birthday for various number of people

the TR ticket becomes a rendezvous node that can reach both the provider and the requester. We will show in the following section that in this rendezvous based propagation, the TR and CT tickets are likely to meet in some rendezvous node in the network. In this case, the trust information can be notified by the rendezvous node and propagated to the requester.

In order to reduce the overhead. Both TR and CT tickets can be piggybacked on the regular packets. Whenever a node sends/forwards a packet, it can include its ticket. As tickets are small, the packet size will not be increased significantly.

3.2 Rendezvous model

Rendezvous point means a meeting point to get-together at a certain time and place. We will prove that the TR and CT tickets can meet at a common rendezvous point in the network. The existence of rendezvous node can be strongly supported by birthday paradox. Birthday paradox describes the probability of some people sharing the same birthday in two randomly chosen small groups [M. 03]. Let us assume there are n number of people, now the probability of at least two people getting same birthday is

$$p(n) = 1 - \frac{365!}{365^n(365-n)!} \tag{1}$$

The plot of birthday paradox for various number of people is shown in Fig. 2. In a group of at least 23 randomly chosen people, there is more than 50% probability that some pair of them will have the same birthday. For 57 or more people, the probability is more than 99%, and it reaches 100% when, the number of people reaches 365 (by the pigeonhole principle).

Birthday paradox is actually a special case of our network when every ticket is query ticket or trust information ticket. Here, date is different participant nodes in the network and the set of people are the nodes receiving either TR ticket or CT ticket. Armed with the birthday paradox concept we now try to find out the probability of obtaining rendezvous node for a given number of nodes, TR tickets and CT tickets.

Suppose there are n birthdays, $r+g$ people with two types of groups (e.g. r women and g men). Each people is randomly assigned by one birthday. Denote the probability that some common birthday exists between the groups as $p(n)$. So the probability that no collision birthday between the group is $1-p(n)$. And,

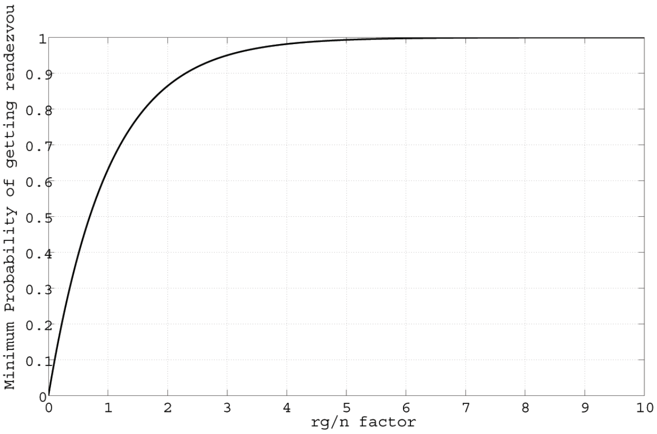


Figure 3 Probability of obtaining rendezvous point for various rg/n factor

$$1 - p(n) = \frac{Number\_of\_no\_collision\_situations}{Number\_of\_all\_possiblesituations} \tag{2}$$

It is obvious that

$$Number\_of\_all\_possiblesituations = 365^{(r+g)} \tag{3}$$

The $Number\_of\_no\_collision\_situations$ is derived as follows.

Consider this situation: if we want to assign i unique birthdays to women's group and j unique birthdays to men's group without collision (i and j are random variables), a two step method can be used: in the first step, we randomly pick $i+j$ unique birthdays. We pick an arbitrary birthday first, then the second birthday can only be one of the remaining 365-1 dates, the third birthday can only have 365-2 dates, etc. That is

$$365 * (365-1) * ... * (365-i-j+1) = \prod_{k=0}^{i+j-1} 365 - k(4)$$

different situations; Then in the second step, we assign i of them to women and j of them to men. This will have S(r,i)*S(g,j) situations (S is Stirling numbers of the second kind).

Since i and j are random variables range in [1,r] and [1,g], the total number of no collision situation will be

$$\sum_{i=1}^{r}\sum_{j=1}^{g} S(r,i)S(g,j) \prod_{k=0}^{i+j-1} 365 - k \tag{5}$$

According to (2)(3)(5), the probability that some common birthday exists between the groups is

$$p(n) = 1 - \frac{1}{365^{r+g}} \sum_{i=1}^{r}\sum_{j=1}^{g} S(r,i)S(g,j) \prod_{k=0}^{i+j-1} 365 - k(6)$$

In general case, when the number of birthday is n,

$$p(n) = 1 - \frac{1}{n^{r+g}} \sum_{i=1}^{r} \sum_{j=1}^{g} S(r,i)S(g,j) \prod_{k=0}^{i+j-1} (n-k) \quad (7)$$

According to reference [J. 10], the formula can be simply approximated by the following equation

$$P(n) \approx 1 - (1 - \frac{1}{n})^{rg} \qquad \approx 1 - e^{-rg/n} \quad (8)$$

Using the birthday paradox concept we can easily find the probability of obtaining rendezvous node. In a wireless distributed network each participant node can be considered as a birthday and each ticket is a person. The CT tickets are analogous to women and the TR tickets are men. The event of node receiving a ticket represents a birthday is assigned to a person. The event of randomly distributing the tickets into the network is similar to the event of people assigned with random birthdays. Assume we have a network of n nodes, and we distribute g TR tickets and r CT tickets uniformly at random. The chance of both the TR tickets and CT tickets cannot meet at a common node is less than $e^{-rg/n}$ [W. 07b, W. 07a]. As long as $rg \geq n$ we have a very high chance of getting a rendezvous node. The probability of obtaining rendezvous node for a given rg/n factor is shown in Fig. 3. From Fig. 3 we can see that the probability of getting a rendezvous point is higher than 0.99 as long as $rg/n \geq 4.61$.

3.3 System model

In the network, each node locally makes trust evaluation on its neighbors, and keeps this direct trust information for a period of time in its buffer. After enough observation time, it will send out CT tickets in the format as: $\{$*Type*, *ProviderID*, *ObjectID*, *TimeStamp*$\}$. This ticket allows the receivers to get the target node's trust value, which is evaluated by the provider. *Type* is a one bit flag, with 0 indicating CT ticket and 1 indicating TR ticket. *TimeStamp* indicates the time when the trust value is computed. For example, after some time of observation, *Jenny* has set up a trust opinion about its neighbor *Lara* and assign a trust value at time t and spread out tickets $\{0$, *Jenny*, *Lara*, $t\}$. When *Jenny*'s another neighbor *Bob* gets tickets, it can query *Jenny* about the trust information on *Lara*. Therefore, *Bob* is qualified to recommend *Lara* to its neighbors by transmitting the ticket.

In the meantime, when a node is interested in the trust value of another node, it will send out TR tickets in format as: $\{$*Type*, *RequesterID*, *ObjectID*, $Time - out\}$. Here $Time - out$ indicates the tolerable delay in receiving the trust information by the requester. For example, if *Alice* is querying for the trust information of *Lara*, it will send out CT ticket in format as: $\{1$, *Alice*, *Lara*, $t'\}$. Assume *Alice* is the neighbor of *Bob*, in this case, *Bob* will also receive the TR ticket. As we described above, as long as $t < t'$, Bob becomes the rendezvous node in this scenario. Therefore the $Time - out$ helps to identify the freshness of the evaluated trust.

3.4 Trust model

Given a requester node r and a target node t, the trust value $T(r,t)$ results in a real number, representing the degree to which the target node should be trusted by the requester node. If the requester node has direct trust evaluation on target node. The requester node use a threshold $\theta(r)$ to determine whether the target is good or malicious. Otherwise a recommender node c is needed, and the trust metric is represented as two real-valued functions, $V(T(r,c), T(c,t), r, c, t)$ and $C(T(r,c), T(c,t), r, c, t)$, the former is the trust value of the second hand evaluation through c and the latter is the confidence of this second hand trust value.

In direct trust evaluation, nodes set up trust relations between each other by evaluating the performance behaviors of direct neighbors. Once a trustworthiness of the node is found by direct trust evaluation, it can be propagated to the network as indirect trust so that trust of nodes which are more than one hop away can be found without recomputations. Direct trust evaluation varies in different applications. For example, node can count its neighbors' good/malicious behaviors and gets statistical 'opinions' about its neighbors. For another example, node can overhear nearby evaluations and then compare them to its local evaluation. If the neighbor evaluations are correlated closely enough with the local evaluation, then the node's evaluation is considered to be valid. The information transmitted by this remote node is considered to be trustful. The detailed description of how trust evaluation algorithms are applied in various distributed applications are out of scope of this paper.

3.5 Attack model

In this paper, we consider following network attack model.

- A malicious node can assign arbitrary trust value to its direct neighbors. However it cannot tamper other node's trust value.
- A malicious node can send bogus trust value as its first hand trust evaluation toward arbitrary node in the network.
- A malicious cannot drop packets when forwarding the packets to other nodes. Otherwise, the sender will recognize it as a malfunctioned node and no longer consider it as a network participant.

This attack model is feasible in distributed ad hoc network with basic security mechanism such as private key encryption.

Next we introduce trust propagation protocol in detail under our trust model and attack model.

4 Trust propagation protocol

In this section, we give the detailed description of propagation protocol. Each node has three buffers: **trust evaluation buffer**, **trust recommendation buffer** and **trust requesting buffer**. Trust evaluation buffer keeps neighbor's trust information by direct observation. This buffer is used to keep track of "*what I have*" for each node. Trust recommendation buffer keeps the provider's ID of the received CT ticket. Trust requesting buffer stores the trust requester's ID of the received TR ticket. Both of them are used to keep track of "*what others (requester or recommender) need from the network*". All the buffers can be updated every other time in order to guarantee a fresh trust value.

In our proposal, each node can distribute two kinds of tickets in the network: TR and CT. The TR ticket $\{1, i, D, t\}$ is used when node i asks for node D's trust information. The requester i only sends out TR query ticket when there is no direct interaction between i and D. After that it waits for the corresponding CT ticket in format as $\{0, R, D, t'\}$ until time out. If i finds a recommender R during the waiting time, (the recommender R either has directly communicated with node D or has the trust information from other recommenders,) a *communication path* will be set up between i and R.

On the other hand, the trust evaluators spread CT tickets after directly evaluating its neighbors. CT ticket is used when a node j volunteers to recommend another node D. It distributes a number of $\{0, j, D, t\}$ tickets indicating that the trust information of node D can be provided by node j. When this ticket is passed to some node R' requesting D's trust information, a *recommendation path* will be set up from j to R'. Based on different roles a node can play in the propagation stage, the specific roles are given as follows:

Ticket Sender: As a TR ticket sender, when a node wants to set up a trust relationship with another node, it checks its *trust evaluation buffer* to find out whether there is a direct trust relationship between them. It returns the target node's trust value if there exists a direct relationship. Otherwise, recommendation is needed to provide trust information. In this case the trust requesting node will distribute the TR tickets into the network containing target node's ID. Then it will wait for the *rendezvoussuccess* ACK message before timeout. If the message is successfully received, requester can calculate the trust metric through the recommendation path. Otherwise, malicious detection fails and another round of rendezvous procedure is needed.

As a CT ticket sender, when a node finishes the direct trust evaluation on other node, it will send out CT tickets. This provides direct trust evaluation for other nodes. Then it will also wait for the *rendezvoussuccess* ACK message before timeout.

The transitivity of the trust value such as concatenation and aggregation have been well studied in trust networks [C-W10] [S. 04]. We will not discuss aggregation and concatenation in this paper.

Algorithm 1: Ticket distribution on node i in trust propagation

```
if $TR$ ticket count $> 1$
    keep one $TR$ ticket at $i$
    distribute additional tickets to $i$'s neighbors
if $CT$ ticket $> 1$
    keep one $CT$ ticket at $i$
    recommend additional tickets to $i$'s neighbors
if $i$ has both $TR$ ticket & $CT$ ticket
    find out the previous node $j$ that send $TR$ to $i$
    send the $CT$ ticket to $j$
```

In order to trace the recommendation path, each node receiving the tickets needs to record its predecessor node. When the RT ticket and the CT ticket meet in rendezvous node, it can calculate the trust value along the recommendation path according to predecessor of the ticket.

Ticket Receiver: Every ticket information is buffered in each intermediate node. Here is how the procedure node i executes when a ticket is received: First, it finds out whether the ticket is a CT or TR ticket by checking the Type flag. If the ticket is a TR ticket, it will check the *ObjectID* to see whether i has direct trust relationship with the object. If the object's trust information is not in its trust evaluation buffer, it will buffer the ticket and distribute the TR tickets to find recommendations for target node. If the received ticket is a CT ticket, it will buffer the ticket and distribute the CT tickets to find trust requester. Then it will check whether it has become a rendezvous node by checking the match between the recommender buffer and the requesting buffer. A match means node i has received both CT and TR tickets for certain trust information. In this case it becomes the rendezvous node and sends the calculated trust value to the requester. The distribution process is shown in Algorithm 1. When the buffer is full, the oldest ticket will be deleted. During the time period a ticket (CT/TR) is valid, if node i receives the counter ticket (TR/CT), then i will become a valid rendezvous node.

When multiple copies of the ticket is received, each node can only keep one copy, it should forward the remaining copies to its neighbors. In our protocol the neighbor is chosen randomly and the remaining tickets are evenly distributed to next hop neighbors.

5 Simulation and Evaluation

We conduct five sets of experiments to evaluate the performance of our approach. In the first set of experiments we examine the malicious node detection rate for various number of malicious nodes. In the second set of experiments, the relationship between the number of tickets and the malicious node detection rate is analyzed. The third set of experiments focus on changing the rendezvous node by running multiple rounds of rendezvous search. In the fourth experiment, we compare the overhead between our method and traditional flooding method. In the last experiment, we evaluate the impact of requester-provider distance.

5.1 Settings

In our simulation set up, 900 nodes are uniformly distributed at random within a rectangular area of $300m \times 300m$ and five of them are malicious nodes. As we assumed the majority of nodes in the network are good nodes, so that the probability of two malicious nodes directly communicating with each other is low. The trust value of a malicious node ranges in (0, 0.1).

All the nodes communicate their TR/CT ticket using UDP datagram. The ticket information is put in the UDP payload. The sender ID and the receiver ID of the tickets are put in the header of the UDP packets. Since the ticket length is very small, when a node has multiple tickets to send, (regarding different nodes' trust value), we can combine them into a single UDP datagram and transmit. The communication range is about $15m$ neglecting the fading effect.

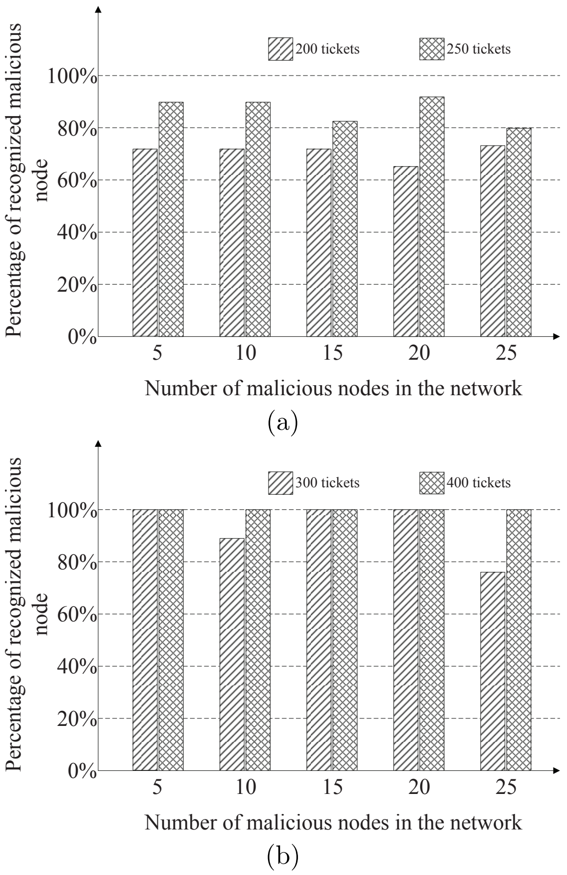
As the bootstrapping phase, each node set up trust opinion of its neighbors by direct trust evaluation. We set ideally good nodes assign higher trust value to good neighboring nodes, and lower trust value to malicious nodes. And a malicious node can assign any trust value to its neighboring nodes.

We use the semiring principle while computing the trustworthiness along the path and between different paths [G. 04]. That is, If there exist multiple rendezvous nodes, and more than one recommendation path is set up between the requester and provider, the requester will pick the highest trust value as its secondary trust value on the object.

We conduct extensive simulation experiments to investigate nodes' trust relationship, and aggregate the percentage of malicious node detection. In every time interval, a group of nodes distribute TR tickets to its neighbors in the network. The trust value is computed along the communication path P_{AB} from trust provider A to rendezvous node r and hence to the information requester B. The trustworthiness of the recommendation path $T(P_{AB})$ is set by the minimum trust value along the path.

Metrics:

For the performance evaluation, we evaluate the performance of our method from two aspects, the accuracy of malicious detection by trust propagation and the overhead of communication in trust propagation. Therefore, we use malicious node detection rate and transmission overhead as metrics. We intentionally distribute malicious nodes randomly and uniformly and evaluate the performance of our proposed trust propagation scheme. We calculate the total packets sent in our method and compare it against the traditional flooding based approach.



(a)

(b)

Figure 4 Probability of recognized malicious node versus the number of malicious nodes using (a) 200, 250 tickets and (b) 300 and 400 tickets

5.2 Results

In the first simulation set up, we increase the number of malicious nodes in the network from five to twenty-five. Fig. 4 shows the percentage of detected malicious nodes versus the number of malicious nodes in the network using our trust propagation scheme. A node will be recognized as malicious node when $T(provider, object) \times T_{P_{pi}}$ is below certain threshold. When the number of malicious nodes is small, most of the malicious nodes can be recognized. The reason is as follows: with more malicious nodes distributed in the network, the probability that malicious nodes exist in the recommendation path increases and hence can effect the trustworthiness of the node it recommended.

In the second part of the simulation, we change the average number of tickets in the network and show its influence on the malicious node detection. The x-axis of Fig. 5 represents the average number of tickets (TR+CT) per query. Result shows that although the recognition accuracy of malicious node is low at first (because of the lack of rendezvous node), it stabilizes at a higher detection rate when the number of ticket reaches a threshold. In the case of 250 tickets, we can get

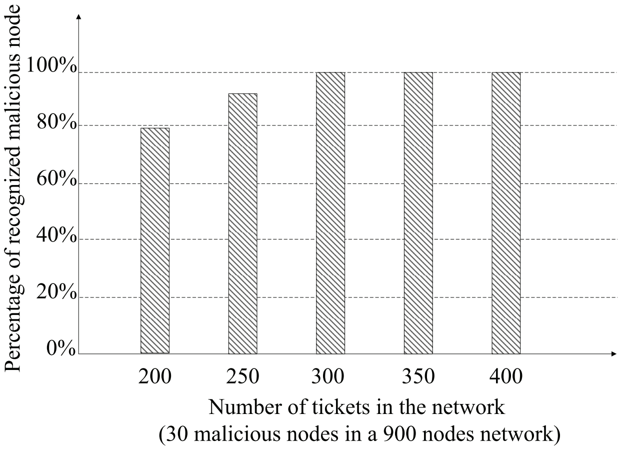
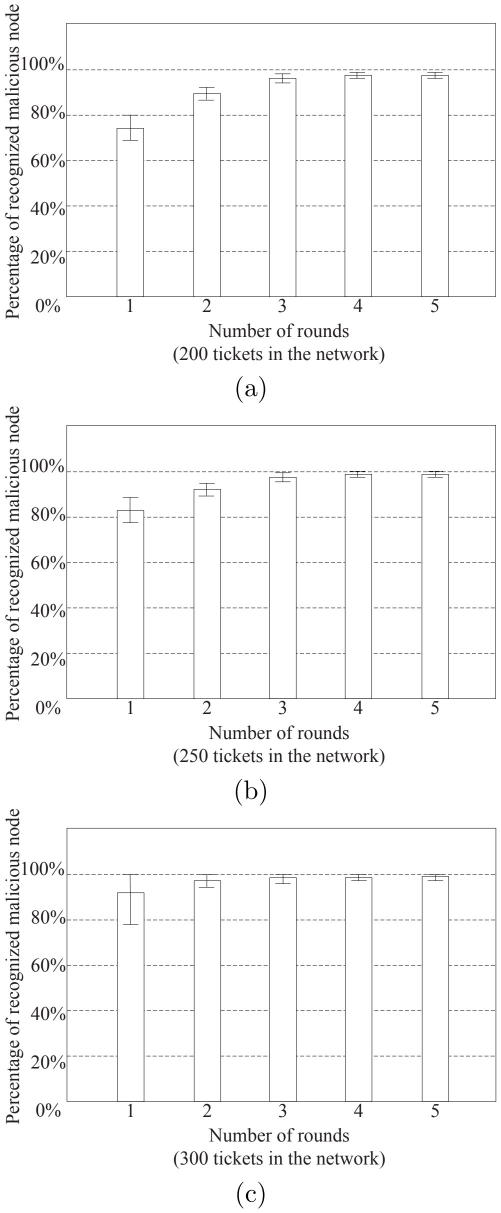


Figure 5 Probability of recognized malicious node versus the number of tickets

about 92% accuracy of recognizing the malicious nodes. Compared to the flooding method, in which we can query all of the 900 nodes to set up all the recommendation paths, although the accuracy of our scheme degrades by 8%, the communication overhead reduces by about 50%. In other words, without compromising performance too much, the communication overhead is reduced in the network.

In the third part of the simulation, we improve the proposed strategy by running multiple rounds of rendezvous search by requester node. Apart from increasing the number of tickets, another way to improve the recognition of malicious node is by multiple rounds of rendezvous discovery. We set the node memoryless in this scenario, that is, the ticket received in the previous round will not be kept for the next round. Fig. 6 illustrate the impact of multiple rounds of rendezvous search on the recognition of malicious nodes. We fix the ticket number at 200, 250 and 300 respectively and run rendezvous algorithms for 1 to 5 rounds. The average percentage of recognition is used to represent the outcome. Results show that the accuracy improvement achieved by increasing the rendezvous search is less than that is achieved by increasing the number of tickets, but still it contributes to the performance improvement. The reason is that when we run multiple rounds of algorithm, we are actually searching for different rendezvous paths and aggregate the results together. This will lead to performance improvement only if new rendezvous node can be found and that occurs when the number of tickets is not too small. Based on observations in both Fig. 5 and Fig. 6, we can conclude that, on one hand ticket number is the most influential factor in deciding the accuracy of our scheme. On the other hand, periodically updating the ticket distribution by running multiple runs will also help to improve the accuracy.

In the fourth experiment, we compare the overhead of traditional flooding based trust propagation and our rendezvous based trust propagation. In the flooding based experiment, each trust message is forwarded to its neighbors with TTL (time to live) set to 10. We



(a)

(b)

(c)

Figure 6 Probability of recognized malicious node versus the number of rounds each node executes

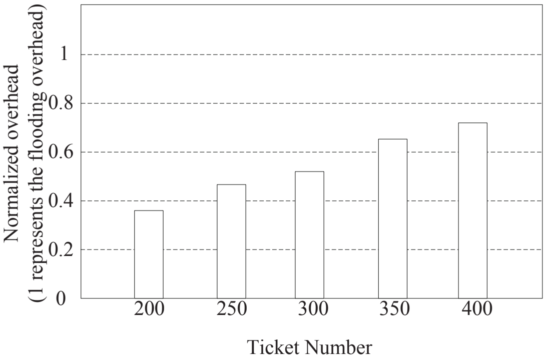


Figure 7 Overhead improvement comparing to traditional flooding method

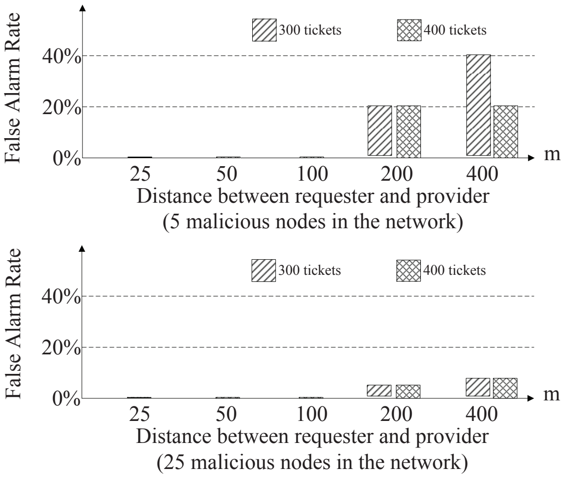


Figure 8 False alarm rate versus the propagation distance

normalize the overhead in flooding based method to 1 as base. We have obtained the normalized overhead of our method for 200, 250, 300, 350, 400 tickets and shown the results in Fig. 7. From Fig. 7 we can see that the overhead reduction we achieve is 66% for 200 tickets and 30% for 400 tickets. Lesser the ticket higher will be the over head reduction. On the flip side less number of tickets reduces performance as shown in Fig. 5.

In each experiment, we examine the number of false alarmed nodes. Our approach works with zero false alarm when the requester and provider is within 200 m. When the distance is over 200 m, few good nodes may be falsely recognized as malicious nodes, due to the long distance discount of trust value. The result is shown in Fig. 8. In the picture, the false alarm rate is the number of false alarmed node divided by the number of malicious nodes. We argue that false alarm is inevitable because of trust discount along the propagation path. In order to minimize it, one of the solutions is to run multiple round of the algorithm to double check the detected malicious node. From this experiment we also conclude that the distance between the requester and the provider will impact the accuracy of malicious detection.

In order to evaluate the impact of requester-provider distance on trust propagation, in this experiment, we vary the number of ticket for different requester-provider pairs regarding their distance (represented by the distance between nodes/network diameter ratio). We change the ticket number and the requester-provider distance. The malicious detection metric is evaluated in Fig. 9.

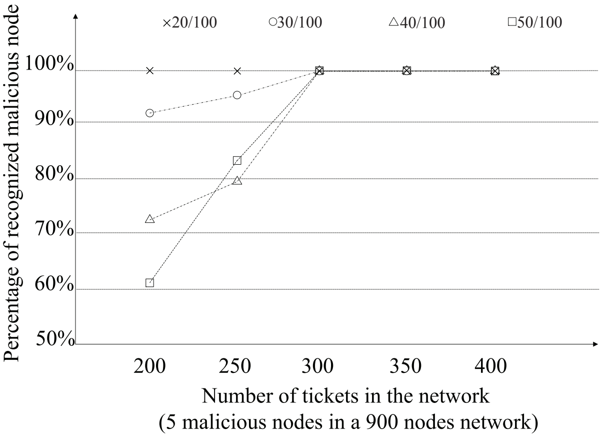


Figure 9 Malicious detection rate under different ticket number and provider-requester distance

Evaluation shows the ticket number can be greatly reduced when the distance between trust requester and trust provider is short. According to this result, instead of using fixed ticket number in trust propagation. We can extend our method by adaptively increasing the ticket number in searching for rendezvous nodes as follows. When the ticket sender does not receive any ACK message before timeout, it may send additional tickets into the network. Since each node can keep at most one copy of the ticket, the additional tickets will be spread into new nodes and increase the existence of rendezvous nodes. In this way, less redundant tickets will be generated during trust propagation.

This paper only discusses trust propagation in statical ad hoc networks. In the next part, we will introduce some properties in more general ad hoc networks and discuss their challenges and advantages for trust propagation, which brings our future research topic in trust propagation.

5.3 Discussions

Trust propagation can be influenced by the network dynamics such as mobility, link stability and network density. Most of the modern networks are highly dynamic in nature and the network compositions keep changing. Table 1 gives the summary of influence of various network dynamics on the trust propagation. It lists both the positive and negative influences. Apart from the promising advantages, trust propagations also has some drawbacks in the over all network performance. The broad summary of influence of trust propagation on the network is given in Table. 2.

Table 1 Influence of various network dynamics on the trust dynamics

Network Dynamics	Trust Dynamics	
	Advantages	Disadvantages
Mobility	Mobility helps to propagate trust naturally [F. 07]. The more mobility the more quicker the propagation of trust.	On the flip side more the mobility larger may be the connection loss and also the neighborhood changes which may impact negatively on the propagation of trust.
Network density	More dense the network is, more faster will be the trust propagation as the dense links make the information flow easier.	More denser the network, the hop length may tends to be shorter as the node can find close by neighbor always. This will eventually increase the number of hops in the communication paths and hence can reduce the ultimate transmitted trust value.
Link stability	More stable the link more trustworthy the trust information transmitted.	Link breakage makes the trust propagation worse. In addition even when the links are strong if the intermediate nodes are misbehaving the ultimately transmitted trust information may not be accurate.

6 Conclusion

We have proposed a rendezvous based trust propagation scheme. The proposed scheme is promising as it reduces overhead and avoids flooding in the network with low false alarm rate. We have analyzed the performance of the proposed schemes for various number of misbehaving nodes and various number of query and trust tickets. The proposed scheme works well despite the presence of misbehaving nodes. The influence of network scales on the proposed trust propagation scheme is also analyzed. This area of research is young and very attractive. There are many issues which have to be addressed including impact of mobility and network dynamics on trust propagation and impact nodes heterogeneity on trust. We hope to address some of these issues in our upcoming research.

Table 2 Influence of trust propagation in the network

Advantages	Disadvantages
(1) Trust computations on node without having direct interactions is possible. Reduces resources spent on recomputation of trust.	Propagation has to be controlled with efficient algorithm otherwise propagation will lead to additional communications over head.
(2) Trust propagation can serve as first level information to prepare a node to have interactions with any strange node.	Propagated trust will never be accurate as it passes through many nodes and may get altered. Direct trust is always accurate as it is node specific and local in nature.
(3) Propagation of trust can help nodes to form a sub group and jointly combat the misbehaving activities.	If the attackers are intelligent enough to launch sybil type of attacks, then the propagation mechanisms may add further confusion in the network.

Acknowledgment

This research was sponsored by the Army Research Laboratory and was accomplished under Cooperative Agreement Number W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official polices, either expressed or implied, of the Army Research Laboratory or the U.S. Government. The U.S Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copy right notation here on.

References

[ABC$^+$08] Reid Andersen, Christian Borgs, Jennifer Chayes, Uriel Feige, Abraham Flaxman, Adam Kalai, Vahab Mirrokni, and Moshe Tennenholtz. Trust-based recommendation systems: an axiomatic approach. In *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pages 199–208, 2008.

[C. 05] C. Chi, X. Sun and Y. Qian. Evaluating the impact of flooding schemes on best-effort traffic. In *Proceedings of IEEE 60th Vehicular Technology Conference*, pages 705–709, Dec.2005.

[C-W09] C-W. Hang, Y. Wang and M. P. Singh. Operators for propagating trust and their evaluation in social networks. In *Proceedings of the 8th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '09, pages 1025–1032, 2009.

[C-W10] C-W. Hang and M. P. Singh. Trust-based recommendation on graph similarity. In *Proceedings of the 13th AAMAS Workshop on Trust in Agent Societies*, May 2010.

[D. 07] D. Quercia, S. Hailes and L. Capra. Lightweight distributed trust propagation. In *In 7th IEEE ICDM*, page 282C291, 2007.

[F. 07] F. Li and J. Wu. Mobility Reduces Uncertainty in MANETs. In *IEEE International Conference on Computer Communications*, 2007.

[F. 08] F. E. Walter, S. Battiston and F. Schweitzer. A model of a trust-based recommendation system on a social network. volume 16, pages 57–74, Feb. 2008.

[G. 04] G. Theodorakopoulos and J. S. Baras. Trust evaluation in ad-hoc networks. In *Proceedings of the 3rd ACM workshop on Wireless security, WiSe 04*, pages 1–10, 2004.

[G-F10] G-F. Liu, Y.Wang, and M. A. Orgun. Quality of trust for social trust path selection in complex social networks. In *Proceedings of the 9th International Conference on Autonomous Agents and Multiagent Systems*, AAMAS '10, pages 1575–1576, 2010.

[H. 04] H. Zhu, F. Bao, R.H. Deng. Computing of trust in wireless networks. In *Proceedings of IEEE 60th Vehicular Technology Conference*, pages 2621–2624, 2004.

[J. 05] J. Golbeck. Computing and Applying Trust in Web-based Social Networks. In *PhD thesis, University of Maryland, College Park*, 2005.

[J. 10] J. Hautakorpi, and G. Schultz. A feasibility study of an arbitrary search in P2P Networks. In *Proceedings of 19th International Conference on Computer Communications and Networks (ICCCN)*, pages 1095–2055, 2010.

[M. 03] M. C. Wendl. Collision Probability Between Sets of Random Variables. In *Statistics and Probability Letter*, volume 64, pages 249–254, 2003.

[N. 11] N. Cheng, K. Govindan and P. Mohapatra. Rendezvous Based Trust Propagation to Enhance Distributed Network Security. In *Proceedings of IEEE INFOCOM workshop on Security in Computers, Networking and Communications*, 2011.

[OS05] John O'Donovan and Barry Smyth. Trust in recommender systems. In *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pages 167–174, 2005.

[R. 04] R. Guha, R. Kumar, P. Raghavan and A. Tomkins. Propagation of trust and distrust. In *In WWW*, page 403C412, 2004.

[S. 03] S. Čapkun, and J. P. Hubaux and L. Buttyán. Mobility helps security in ad hoc networks. In *Proceedings of the 4th ACM international symposium on Mobile ad hoc networking, Mobihoc 2003*, pages 46–56, 2003.

[S. 04] 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.

[S. 05] S. M. Das, H. Pucha and C. Y. Hu. Performance comparison of scalable location services for geographic ad hoc routing. In *Proceedings of IEEE International Conference on Computer Communications(INFOCOM)*, 2005.

[S. 06] S. Flowerday and R.V. Solms. Trust an element of information security . In *In Security and Privacy in Dynamic Environments. IFIP/SEC2006*, pages 87–97, 2006.

[U. 07] U. Kuter and J. Golbeck. Sunny: A new algorithm for trust inference in social networks using probabilistic con?dence models. In *In AAAI*, page 1377C1382, 2007.

[W. 07a] W. W. Terpstra, C. Leng and A. P. Buchmann. BubbleStorm: Analysis of Probabilistic Exhaustive Search in a Heterogeneous Peer-to-Peer System. In *Technical Report TUD-CS-2007-2 Technische Universitt Darmstadt, Germany*, 2007.

[W. 07b] W. W. Terpstra, J. Kangasharju, C. Leng and A. P. Buchmann. BubbleStorm: Resilient, probabilistic and exhaustive. In *P2P search, Proc. ACM SIGCOMM*, pages 49–60, 2007.

[Y. 05] Y. Sun, W. Yu, A. Han and K. J. R Liu. Trust Modeling and Evaluation in Ad Hoc Networks. In *Proceedings of the IEEE Global Telecommunicationtions Conference, Globecom'05*, pages 1862–1867, 2005.

[ZL04] Cai-Nicolas Ziegler and Georg Lausen. Spreading activation models for trust propagation. In *EEE '04: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)*, pages 83–97, 2004.

[ZL05] Cai-Nicolas Ziegler and Georg Lausen. Propagation models for trust and distrust in social networks. *Information Systems Frontiers*, 7(4-5):337–358, 2005.