Rendezvous Based Trust Propagation to Enhance Distributed Network Security

Ningning Cheng, Kannan Govindan and Prasant Mohapatra
Department of Computer Science
University of California, Davis, CA 95616
Email: {chengni, gkannan, 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.**

I. 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 [1]. 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 [2], [3]. 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.

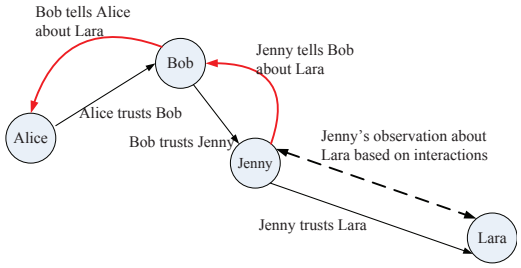

Fig. 1. Propagation of trust in a simple straight chain

Both the overhead cost and propagation delay 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*.

For instance in Fig. 1 requester $Alice$ wants to know the trustworthiness of target $Lara$; however $Lara$ is out of its communication range. $Jenny$ who is neighbor of $Lara$ has direct trust relationship with $Lara$, and 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.

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 bad nodes can report dishonest message and be uncooperative and selfish.
- The majority of the network member is good. There are no consecutive bad nodes along a single communication path.
- There is no collusion attacks in the network, which means all the bad 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 II reviews some of the related work in the trust propagations. Foundation for the proposed trust propagation scheme is provided in Section III. Our proposed trust propagation model is presented in detail in Section IV. Performance evaluation of the proposed scheme is carried out in Section V. Concluding remarks are given in Section VI.

II. Related work

Trust propagation in small world network is proposed in [4]. In this network when the nodes form a trust propagation path,

978-1-4244-9920-5/11/$26.00 ©2011 IEEE

it is relatively short due to the small world influence. This approach can be used only in certain specified self-organized ad hoc network.

Trust propagation in social networks have been studied in [5]–[10]. These work focus more on trust concatenation, aggregation and path selection in a social semantic web graph, which differs from wireless communication, where 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 [11]–[13]. 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 [14], where nodes exchange trust information as soon as they are connected. The 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 [15], [16]. Spreading activation is a method for searching trust values or any intended values of nodes in the networks.

III. Foundation of the propagation model

A. Terminology

In order to give a formal description of trust and trustworthiness, we give our definitions of trust/trustworthiness in the paper as follows:

DEFINITION 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 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: **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 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.

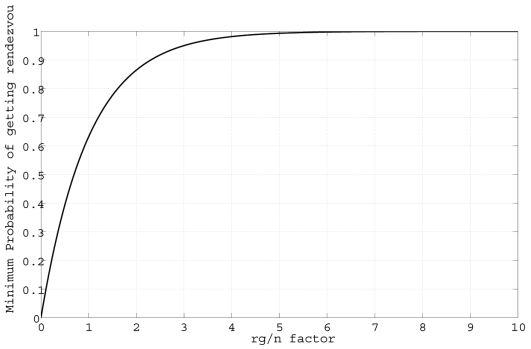

Fig. 2. Probability of obtaining rendezvous point for various rg/n factor

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.

B. Rendezvous model

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 [17].

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. Now the probability that some common birthday exists between the groups is

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

where S is Stirling numbers of the second kind. And the formula can be simply approximated [18] by the following equation

$$P(n) \approx 1 - (1 - \frac{1}{n})^{rg} \approx 1 - e^{-rg/n} \quad (2)$$

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 $e^{-rg/n}$ [19], [20]. 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. 2. From Fig. 2 we can see that the probability of getting a rendezvous point is higher than 0.99 as long as $rg/n \geq 4.61$.

C. 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.

D. Direct trust evaluation

In a trust secured network, nodes set up trust relations between each other by evaluating the performance behaviors of direct neighbors. This stage is called direct trust evaluation. 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/bad 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.

IV. 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. It searches for a recommender by TR query ticket when there is no direct interaction between i and D. Then it waits for the corresponding CT ticket in format as $\{0, R, D, t',\}$ until time out. After i finds a recommender R, which 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. 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 play in the propagation stage, the specific roles are given as follows:

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.

When a node finishes the direct trust evaluation on other node, it can send out CT tickets. This provides direct trust evaluation for other nodes. The transitivity of the trust value such as concatenation and aggregation have been well studied in trust networks [21] [22]. 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
```

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.

V. Simulation and Evaluation

We conduct four 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 end, we compare the overhead between our method and traditional flooding method.

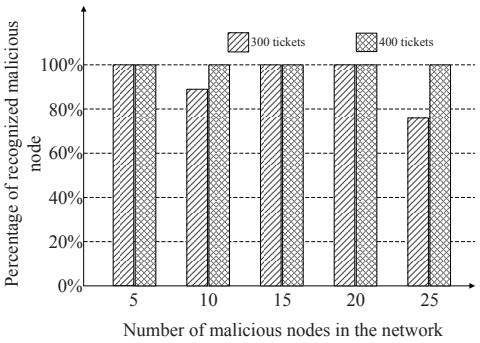

Fig. 3. Probability of recognized malicious node versus the number of malicious nodes

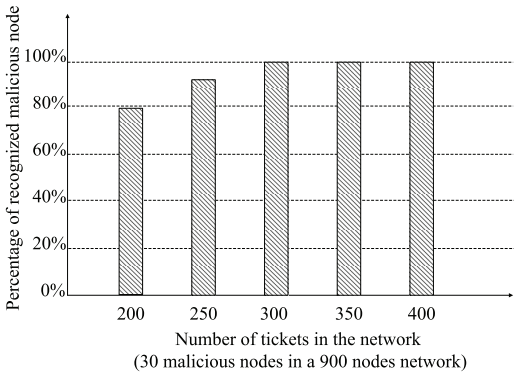

Fig. 4. Probability of recognized malicious node versus the number of tickets

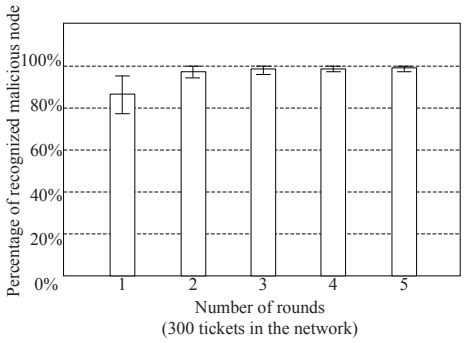

Fig. 5. Probability of recognized malicious node versus the number of rounds each node executes

A. 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. The communication range is about $15m$ neglecting the fading effect. We assume the semiring principle while computing the trustworthiness along the path and between different paths [23].

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 use malicious node detection rate and communication 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.

B. Results

In the first simulation set up, we increase the number of malicious nodes in the network. Fig. 3 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 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. 4 represents the average number of tickets (TR+CT) per query. It 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 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. Fig. 5 illustrate the impact of multiple rounds of rendezvous search on the recognition of bad nodes. We fix the ticket number at 150 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. 4 and Fig. 5, 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 versus our rendezvous based trust propagation by one round. In the flooding based

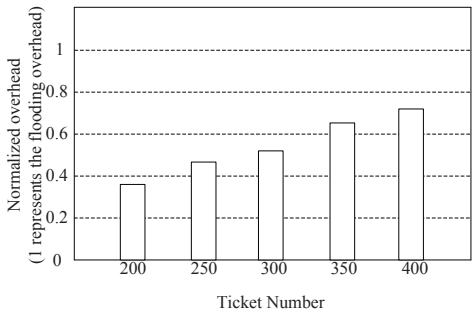

Fig. 6. Overhead improvement comparing to traditional flooding method

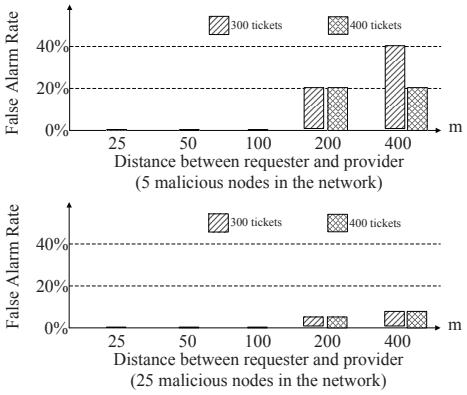

Fig. 7. False alarm rate versus the propagation distance

experiment, each trust message is forwarded to its neighbors with TTL (time to live) set to 10. We 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. 6. From Fig. 6 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. 4.

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. 7. 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.

VI. 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.

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

[1] S. Flowerday and R.V. Solms, "Trust an element of information security ," in *In Security and Privacy in Dynamic Environments. IFIP/SEC2006*, pp. 87–97, 2006.
[2] 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*, pp. 705–709, Dec.2005.
[3] 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 Telecommuniactiontions Conference, Globecom'05*, pp. 1862–1867, 2005.
[4] H. Zhu, F. Bao, R.H. Deng, "Computing of trust in wireless networks," in *Proceedings of IEEE 60th Vehicular Technology Conference*, pp. 2621–2624, 2004.
[5] 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, pp. 1575–1576, 2010.
[6] 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, pp. 1025–1032, 2009.
[7] U. Kuter and J. Golbeck, "Sunny: A new algorithm for trust inference in social networks using probabilistic con?dence models," in *In AAAI*, p. 1377C1382, 2007.
[8] D. Quercia, S. Hailes and L. Capra, "Lightweight distributed trust propagation," in *In 7th IEEE ICDM*, p. 282C291, 2007.
[9] R. Guha, R. Kumar, P. Raghavan and A. Tomkins, "Propagation of trust and distrust," in *In WWW*, p. 403C412, 2004.
[10] J. Golbeck, "Computing and Applying Trust in Web-based Social Networks," in *PhD thesis, University of Maryland, College Park*, 2005.
[11] F. E. Walter, S. Battiston and F. Schweitzer, "A model of a trust-based recommendation system on a social network," vol. 16, pp. 57–74, Feb. 2008.
[12] R. Andersen, C. Borgs, J. Chayes, U. Feige, A. Flaxman, A. Kalai, V. Mirrokni, and M. Tennenholtz, "Trust-based recommendation systems: an axiomatic approach," in *WWW '08: Proceeding of the 17th international conference on World Wide Web*, pp. 199–208, 2008.
[13] J. O'Donovan and B. Smyth, "Trust in recommender systems," in *IUI '05: Proceedings of the 10th international conference on Intelligent user interfaces*, pp. 167–174, 2005.
[14] 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*, pp. 46–56, 2003.
[15] C.-N. Ziegler and G. 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)*, pp. 83–97, 2004.
[16] C.-N. Ziegler and G. Lausen, "Propagation models for trust and distrust in social networks," *Information Systems Frontiers*, vol. 7, no. 4-5, pp. 337–358, 2005.
[17] M. C. Wendl, "Collision Probability Between Sets of Random Variables," in *Statistics and Probability Letter*, vol. 64, pp. 249–254, 2003.
[18] 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)*, pp. 1095–2055, 2010.
[19] W. W. Terpstra, J. Kangasharju, C. Leng and A. P. Buchmann, "BubbleStorm: Resilient, probabilistic and exhaustive," in *P2P search, Proc. ACM SIGCOMM*, pp. 49–60, 2007.
[20] 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.
[21] 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.
[22] 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.
[23] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proceedings of the 3rd ACM workshop on Wireless security, WiSe 04*, pp. 1–10, 2004.