ITEC 810 Final Report:

A Survey of Web-based Social Network Trust

Student:
Eric Wang
41176774

Supervisor:
Yan Wang

Subject Coordinator:
Robert Dale

Abstract

Trust has become more important in Web-based Social Networks (WBSNs) where trust relation between users is used to seek reliable information. Furthermore, the presence of malicious attack in providing false information is increasing in WBSN. In existing WBSN platforms, a user specifies trust values to the parties which he has a direct relationship with. This user then makes a decision based on the trust value of those parties and their opinions. However, when the user seeks information from parties who are not directly connected to him, then he needs to determine the reliability of this information and also ensures that it is not from a malicious user by using trust inference. The aim of this paper is to present an overview of existing trust inference mechanisms in WBSN that are designed on solving this problem. Analysis and comparison are then performed to propose issues for future studies.

Acknowledgements

The author would like to thank the assistance of his supervisor Dr. Yan Wang and also the PhD student Guanfeng Liu in their tireless efforts in reviewing and correcting this paper on numerous occasions. Their valuable suggestions are greatly appreciated.

Table of Content

Abstract 2
Acknowledgements 3
1 Introduction 5
2 Trust Definition and Characteristics 7
3 Trust Inference Mechanisms 11
3.1 TidalTrust 11
3.2 Binary Trust Algorithms and TrustMail 13
3.3 Advogato Trust Metric 14
3.4 Appleseed 15
3.5 SocialTrust 16
3.6 FuzzyTrust Algorithm 17
3.7 SUNNY 18
3.8 Trusted Gossip 19
3.9 RN-Trust 20
4 Findings and Discussion 22
4.1 Approach of the Problem 22
4.2 Merits and Weaknesses 22
4.3 Possible future studies 23
5 Conclusion 25

1 Introduction

In a Web-based Social Network (WBSN), such as Facebook, MySpace, LinkedIn and LiveJournal, information is created and consumed by its users. Users share information based on the level of trust they explicitly assign to other users who are directly connected to them. Thus, a user can determine the reliability of a piece of information from another user by the trust value he assigns to the user who is the source of the information.

The ability to determine how much a user should trust the source of the information when the user does not know the source directly can be used for aggregating, filtering, and ordering of information. Additionally, if this indirect trust, or trust inference, can be estimated accurately with minimum negative impacts from malicious users who provide false information, then the user can use this trust estimation to make decisions on the reliability of the information.

In another word, the problem we face with trust in WBSNs is the ability to determine how much one user should trust another user who is not directly connected to him/her in the same WBSN. It is beneficial to automate the process of finding trusted user who is not directly connected with another user by trust inference estimation so that a user can seek trusted information outside his direct contacts and determine the reliability of this information via his trusted contacts.

For trust estimation to be useful in WBSN, it is often expressed as trust ratings or values that a user can explicitly assign to another user. We can then use those trust values to infer the trust that may exist between two people who are not directly connected. In another word, when trust is explicitly rated on a numerical scale, this network data can be composed to produce information about the trust between two individuals without a direct connection. [Jennifer Golbeck and James Hendler, 2006].

Trust inference estimation must be simulated into an efficient and accurate mathematical algorithm, or mechanism, to calculate the trust inference value. The structure of the social network and explicit trust value between two directly connected users can be used in this calculation. Further more, this mechanism needs to be effective against malice attacks from users who give false trust values to benefit themselves.

This trust inference value can enable users to make reasonable decisions on the basis of the trust relationship in WBSN with certainty and degree of confidence. For example, a user may want to know if a certain book is worth reading. He may need to seek the opinions from users he is not directly connected to due to the fact that none of the users who he is directly connected to has read the book. He needs to know how reliable those opinions are via the users he is directly connected with, to the users who are directly connected to users he is directly connected to, but are not directly connected to himself. In another word, user A may need to seek an opinion of a book from user B who user A knows directly. User B does not have this information because he has not read this book but he knows user C who user B knows directly has this opinion. Then user A needs to determine how reliable this opinion is from user C via his direct connection with user B.

A great example of how trust inference estimation can be beneficial where there may be conflicting information from different sources of information is given in the work by Mohsen Lesani *et al* on FuzzyTrust. This is quote below.

People read the books from a writer if they themselves know the writer or some one that they trust in reading stuff introduces the writer to you. Similarly you listen to music by a composer you know or that is introduced by some one you believe in music. Other people recommendations make us to do things many times such as watching a film, reading a paper and so on. Trust may seem not to be transitive i.e. if A highly trusts B and B highly trusts C does it not mean that A also highly trusts C? It is usual that you ask one of your highly trusted friends about an unknown and take his opinion as your own. Consider a case when A asks B about a film because A highly trusts B and B does not know about the film so B asks her or his highly trusted friend C that knows about the film while A is unaware of this relationship. B takes C's opinion as her or his own and gives it to A that will also take it as her or his own. A finally takes C's opinion. It is seen that trust is transitive in this sense. Transitivity lets the trust to a person to pass back through a chain of people. Now consider if A does not know C, she or he asks her or his friends (Bs) about C. Different friends (Bs) may have different ideas about that person (C). Person A should compose the different ideas she or he receives about C from Bs to infer a unique idea about C. People naturally compose trust value when they receive them from different sources maybe giving higher importance to more trusted sources. The conflicting information may be present and composing strategies are desired to handle integration of such information. Transitivity and composability are what make the trust graph in a network to have more trust information than is explicitly specified in the graph edges. Modeling the persons as nodes and friendships or acquaintances as directed edges and trust values as edge labels, the theoretical problem is to infer trust from any node in the directed graph to any other. [Mohsen Lesani and Saeed Bagheri, 2006]

The rest of the paper is organised as follows; in section 2, we discuss what trust is and what trust inference is in the context of WBSN. We also illustrate the concept of trust inference by the use of trust diagram. In section 3, analysis and comparisons are performed on a collection of trust inference mechanisms designed for determining trust inference value. Then in section 4, we discuss the findings of this paper. Lastly in section 5 we conclude this paper by presenting some issues for future studies based from our findings.

2 Trust Definition and Characteristics

The verb trust, in general terms, can mean to have belief or confidence in the honesty, goodness, skill or safety of a person, organization or thing [Cambridge Dictionaries Online, 2009], or simply to have confidence or faith in a person or a piece of information [Jennifer Golbeck., 2005].

Some social network sites have trust evaluation model applied in their network structure. For instance, “LinkedIn connects you to your trusted contacts and helps you exchange knowledge, ideas, and opportunities with a broader network of professionals” [learn.linkedin.com/what-is-linkedin/].

To simulate trust in the WBSN environment, J. Golbeck and J. Hendler [Jennifer Golbeck, James Hendler, Nov 2006], proposed that there are three main properties of trust. They are asymmetry, personalisation, and transitivity. It is often useful to represent these properties as a trust diagram when we describe trust inference.

The *first property*, trust asymmetry, means trust is not necessarily the same in both directions between two users. For example, Alan trusts his manager Billy, but Billy may not trust Alan with the same amount of trust. This is shown as a directed arrow from one user or node to another in a trust diagram to indicate the direction of trust so that we know we are refering to the trust from Alan to Bılly, or from Billy to Alan.

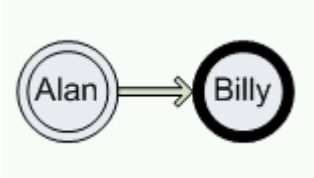

Figure 1: Alan trusts Billy

The *second property* is trust personalisation, where it is suggested that inherently trust is a personal opinion. For instance, Alan and Billy may have very different opinion about Clark; however, there is no absolute correct or incorrect value except from the perspective of Alan and Billy. This is in contrast to a reputation system such as in a P2P network where there is a global value of trust on a particular user or node or. This trust personalisation property is shown as different trust values (0.9 from Alan to Clark, and 0.4 from Billy to Clark, where 1.0 is full trust and 0.0 is no trust) from different users to a single user in a trust diagram.

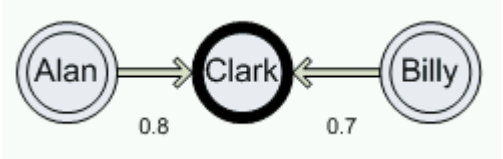

Figure 2: Alan and Billy both trusts Clark but at different degree of trust

The *third property* of trust that they proposed is that trust is not perfectly transitive in the mathematical sense where if Alan trusts Billy, and Billy trusts Clark, then it is not necessarily true for Alan to trust Clark in the same level of

trust that Billy trusts Clark. However, Alan does trust Clark via Billy. It is this transitivity of trust that allows indirect trust, or trust inference to pass from Alan to Clark via Billy. This property allows trust inference value to be calculated from one user or node to anther via other nodes in between.

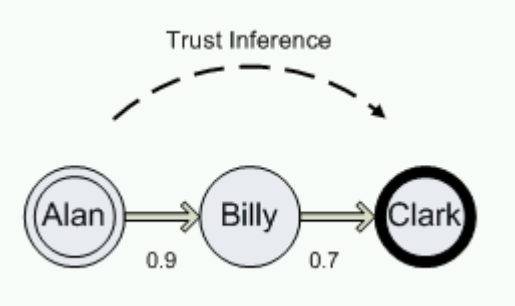

Figure 3: Trust Inference from Alan to Clark

Furthermore, there are two aspects of this transitivity of trust, where there is the trust in a person, and then there is the trust in the person's recommendations of other people. For example, Alan may trust Billy's opinion on music, but not trust him on recommending on other people on their opinions about music. Often the trust in person is used for trust inference in a WBSN, while trust in opinion or performance is used for trust reputation in a P2P network.

In decribing the of trust inference in a WBSN, a trust diagram is often used. This is explained in the following paragraphs.

In Figure 1, we show the definition of direct trust or explicit trust in WBSN. A person is represented by a node, while a trust is represented by a direction arrow. We define node A being the source of the trust where it is directly connected to Z and has a direct trust to node Z by the directed arrow. We say node Z is the sink of the rust where Z is being trusted by A.

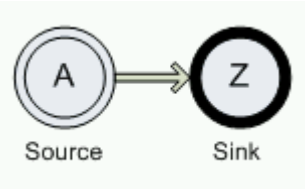

Figure 4: Direct or Explicit Trust

In Figure 2, we illustrate indirect trust, or trust inference in WBSN. Here we also define node A being the source of the trust where it is directly connected to B. Node B is directly connected to node C. However, node A is not directly connected with node C. If node A wants to trust C, then we call this trust as an indirect trust, or trust inference of node A to node C, where C is the sink of the trust. The trust rating of this indirect trust, or trust inference value for which we shall call in this paper, is defined as the trust value of node A to node C, as if there is a direct trust from node A to node C.

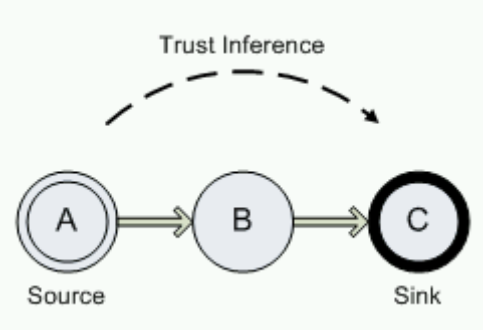

Figure 5: Indirect trust, or Trust inference

In Figure 3, we further demonstrate what trust inference is in a trust diagram. Again, we define node A, or the source node, is directly connected with B and C, but is not directed connected to D, E, F, or G. Furthermore, we show that A is indirectly connected to G via four paths, A->B->D->E->G, A->C->D->F->G, A->B->D->F->G, and A->C->D->E->G, thus creating four perspectives or trust inference values of G, or the "sink", when we are determining the trust inference value of A to G.

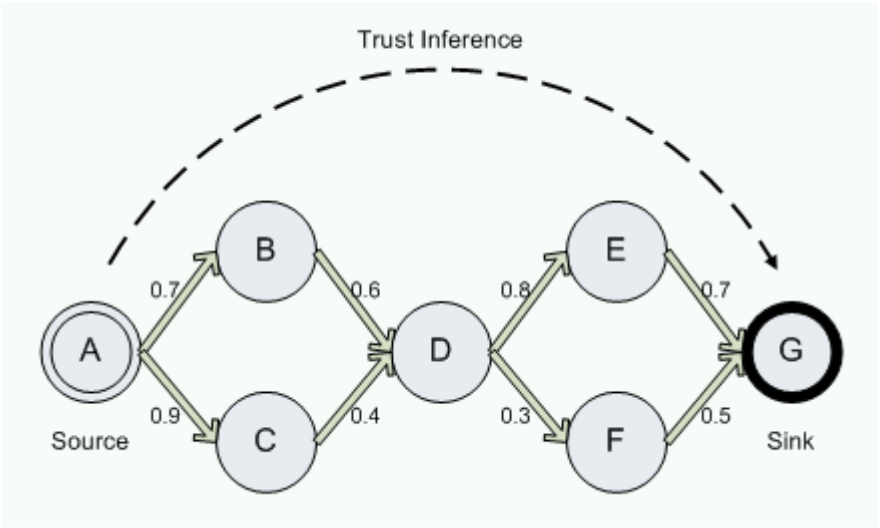

Figure 6: Trust inference from node A to node G

A trust inference mechanism can be decribed as an algorithm which determines the inference trust values which are recommendations to one user about how much to trust another user. This algorithm can generate a recommended trust rating for an unknown person in the social network, based on information from others user connecting to this unknown user.

Modeling a trust inference mechanism is achieved by assigning the persons in a WBSN as nodes and friendships or relationships as directed edges, while trust values as edge labels.

If someone a user (the source) wants to know how much to trust another user (the sink), we can look at the who trusts the sink along the path and see how much the source trusts the intermediate people, then produce a result that weights trust inference value from trusted people who are more highly than those from less trusted people.

We can see trust is subjective from the above properties and that there can be no single global value of trust because each trust from each individual is different. The main difference between trust in P2P networks and WBSNs is that in P2P networks, trust is a measure of performance, and one would not expect the performance of user A to be very different when it is interacting with user B compare with user C. Therefore, in P2P networks, a single global recommendation about the trustworthiness of a user will usually be sufficient. However, in WBSNs, two users (B and C) can have entirely different opinions about the trustworthiness of the same user A.

A characteristic of trust inference estimation in WBSN is the issue of malicious users spreading false information for their own gains. Jesse Ruderman illustrate this problem and suggested trust inference algorithm has to be attack-resistant to minimise the impact from this problem. His descritpion is quoted below.

While there is no commonly agreed definition of what it means for a trust metric to be attack resistant, we can think of it as meaning that the attacker's success is bounded in some way by the number of confused nodes or some other property of the confused nodes. An attacker cannot defeat an attackresistant trust network simply by creating a huge number of pseudonyms and connecting them in the right way; he must cause existing users to become "confused" and certify his nodes. An attack-resistant trust network must have a trusted seed. Without one, an attacker could create a pseudonym corresponding to each existing user, and set up an identical certificaiton graph so the trust network cannot tell the difference. An example of a trust metric that is attack-resistant but not useful is one that simply returns the seed. It is attack-resistant because bad nodes are not trusted at all, but it is not useful because it puts all of the burden of deciding trust on the choice of the seed. An example of a trust metric that is not attack-resistant is one that accept all nodes within distance k from the seed.. It is not attack-resistant because once an attacker convinces a "confused" node with distance k−2 or less to certify a bad node, the bad node can certify as many bad nodes as it wants, and those nodes will all be within distance k from the seed. [Jesse Ruderman, 2004]

3 Trust Inference Mechanisms

There are many trust inference mechanisms proposed by scholars around the world, showing the popularity of this topic. However, at the same time, this indicates the topic is yet to mature and converge into industry standards from various existing mechanisms.

The following sub sections (3.1 to 3.9) are a collection of trust inference mechanisms designed for WBSNs. Some are algorithms; others are models that can be more complex.

Trust inference mechanisms test their accuracy by taking an existing social network data set, and then arbitrarily take away selective number of direct links. Then trust inference calculation is performed on those links, and finally the result of their trust inference value and the explicit trust value of the direct links are compared to determine the accuracy of the trust inference mechanism.

3.1 TidalTrust

The TidalTrust algorithm is proposed by Golbeck [Jennifer Golbeck, 2005]. It considers the trust values to be numbers in a continuous range of [0...10]. It is simple and its low complexity (O(V+E)) allows high scalability in its application.

Let t_{ij} represent the trust rating from user i to user j. Let t_{js} represent the trust rating from user j to user s. The inferred trust rating from user i to user s via user j is given by below.

$$t_{is} = \frac{\sum_{j \in adj(i)} t_{ij} t_{js}}{\sum_{j \in adj(i)} t_{ij}}$$

Equation 1: TidalTrust Algorithm

TidalTrust was named because its calculations sweep forward from source to sink in the network, and then pull back from the sink to return the final value to the source.

Mohsen Taherian, et al describe Golbeck's TidalTrust algorithm as quoted below.

TidalTrust algorithm considers the trust values to be numbers in a continuous range from 0 to 10. As mentioned before, a directed graph is used to represent trust relationships. Each edge has a label in the range from 0 to 10 which 10 means full trust and 0 means no information. Suppose that the node s in the trust network wants to compute its trust value to the node t which is not directly connected to s. First, s sends a request to all its neighbors. This request is recursively forwarded until it reaches to nodes

having an edge to t. Then, the trust values of these nodes to t are moved backward across the paths that their corresponding requests are came from. In the backward path, when a node receives more than one value, it uses WAO (weighted average operator) to combine these trust values. This scenario continues until trust values reach to s across the same paths of sending requests but in reverse direction. After this, s usesWAO operator like the other nodes to combine its received trust values and compute final trust value from s to t. [Mohsen Taherian, et al, 2008]

Perhaps, the most important preference of Golbeck's algorithm with respect to other ones is its simplicity and its low time complexity (O(V+E)). Although she has used a simple operator to combine trust values, looking at the results of other algorithms, we can see that Golbeck's results are better in many cases. This is mainly because of her applied restrictions in the algorithm. TidalTrust algorithm is known as a famous and highly cited algorithm for inferring trust. This is why we chose TidalTrust algorithm as the basis of our results comparison. Despite of popularity of the TidalTrust algorithm, it has some problems. First, we must mention that with Golbeck's restriction on the length of inference paths, some useful information may be missed. Yet, the more important problem is about single paths (chains) between the source (s) and the sink (t). Suppose that there is only a long chain between the source and the sink, and all nodes in this chain have trust value 9 (high trust) to their neighbors. Now, suppose that there is another chain with the same length with this difference that all nodes have trust value 10 (full trust) to their neighbors except the one just before the sink which has trust value 9 to the sink. With the TidalTrust algorithm there is no difference between these two cases. In both cases, the computed trust value from s to t is 9. However, it is clear that in the former case the inferred trust value must be smaller than the latter one. In fact, increasing the chain length should have a reverse relation with the final inferred trust. [Mohsen Taherian, et al, 2008]

Golbeck assumes trust values inferred through shorter paths may be more accurate, thus only the shortest paths from source to sink are considered. This works for the algorithm because it simplified the algorithm; however, its weakness is it excluded information that may be useful from nodes of longer paths, especially the case where majority of the nodes in the longer path may have more trusted nodes.

One important factor in applying TidalTrust is that each WBSN is different. Depending on the topic of which trust is being determined, the user community, and the structure of the network, the impact of different trust properties can differ greatly. While we should still expect the general principles to be the same where shorter paths will be more accurate than longer ones, and also that higher trusted people will agree with us more than less trusted people, the proportions of those relationships may be different from a WBSN used in a given research.

Two conditions were applied on Globeck's algorithm after she examined the structure of real world WBSNs and their properties. Firstly, she showed that trust inference values through shorter path is often more accurate, therefore she only considered the shortest path from source to sink in her trust inference algorithm. Secondly, she extracted from her analysis that the most

trustworthy information often comes from highest trusted neighbours and lower trusted neighbours give lower trustworthy information. From this, she deduced that she can limit a trust threshold for trust network in her algorithm and applied this threshold in combining trust values. She showed that these restrictions lead to more accurate results in many cases.

Even though the TidalTrust algorithm is very popular, it has two major problems. First, Golbeck's restriction on the length of inference paths means some useful information may be missed. Yet, the more important problem is about single paths between the source and the sink. Let us suppose that there is only a long chain between the source and the sink, and all nodes in this chain have trust value 9 (high trust) to their neighbors. Now, suppose that there is another chain with the same length with this difference that all nodes have trust value 10 (full trust) to their neighbors except the one just before the sink which has trust value 9 to the sink. With the TidalTrust algorithm there is no difference between these two cases. In both cases, the computed trust value from s to t is 9. However, it is clear that in the former case the inferred trust value must be smaller than the latter one. In fact, increasing the chain length should have a reverse relation with the final inferred trust. [Raphael L. Levien, 2002]

TidalTrust algorithm is known as a famous and highly cited algorithm for inferring trust. It is a simple yet generic algorithm in that it can be applied to any WBSN trust inference calculation. Many other trust inference algorithms found in this paper make comparison of their trust algorithms with TidalTrust.

3.2 Binary Trust Algorithms and TrustMail

Two similar algorithms were proposed [Jennifer Golbeck, and James Hendler, 2006] which aimed to develop efficient and accurate mechanisms for inferring trust relationships that use only the structure and trust ratings within a social network. They integrate social network structure and trust ratings based on binary trust assignment (0 or 1) and input the results into an experimental email client application called TrustMail to enhance email filtering.

The Binary Trust algorithms are the Rounding Algorithm and the Non-Rounding Algorithm. Both of them use binary value (0 for not trustworthy while 1 is trustworthy) to explicitly indicate trust value assign at each node. Both Rounding Algorithm and Non-Rounding Algorithm give similar results but Rounding Algorithm is more accurate than non-rounding at each node of the calculation, when over half of the nodes are "good nodes" that provide true trust values most of the time. This is because rounding at each node removes more error than only performing the rounding at the last node.

The Rounding Algorithm and the Non-Rounding Algorithm both calculate a result based on the percentage of each node's adjacent nodes that rates the node as good (value of 1) or bad (value of 0). Their experimental results show increase in bad nodes does not significantly reduce the accuracy of the results. Thus both algorithms are robust against malicious attacks from bad users, or "bad nodes" which intentionally provide false trust values.

Their experimental email client software, TrustMail, can replace the manual process of verifying the trustworthiness of the information about an email

sender/recipient by utilising the data in WBSN, thus assists in filtering emails and can work in conjunction with spam filtering software. This works by the user rates an email recipient that the user knows directly, and each email received in the inbox will be rated by a trust inference value calculated by their binary trust algorithm. This is to achieve higher ratings to non-spam senders.

The researchers admit that this filtering method does not replace the currently used agent based approach such as blacklisting and white listing email filtering, but can act a complementary system to ease email overload on users.

3.3 Advogato Trust Metric

This trust mechanism [Raph Levien, 2002] calculates trust by using a network flow model. It issues 3 levels of certification between users to determine the user trust value within a group.

Advogato uses the same trusted nodes, or seeds, to determine trust calculation for all users, thus it is similar to a global trust algorithm that is suitable for P2P. However, a common modification is to set the user as the trusted node, therefore transforming it to a local trust algorithm suitable for WBSN.

Advogato calculates the inferred trust value by identifying individual bad nodes and finding any nodes that certify the bad nodes, the metric eliminates the unreliable section of the network, making it more resistant to malicious users.

Jesse Ruderman [Jesse Ruderman, 2004] describes the Advogato trust algorithm as quoted below.

“Advogato’s trust metric is based on network flow. First, the Advogato metric assigns a “capacity” cx to each node x as a non increasing function of the distance from the seed. For example, advogato.org uses a capacity of 800 for the seed, 200 for the next two levels, 50 for nodes with distance 3 from the seed, and so on. Each node A is then split into two parts A− and A+, with a capacity-1 edge from A− to the sink and a capacity-(cx −1) edge from A− to A+. A’s certification of B becomes an infinite-capacity edge from A+ to B−. Advogato uses the Ford-Fulkerson algorithm to find the maximum flow. Since Ford-Fulkerson always picks the shortest augmenting path from the seed, any node with flow from x− to x+ also has flow from x− to the sink. Ford-Fulkerson takes O(|f ||E|), where f is the maximum flow. In this graph, |f | is simply the number of nodes accepted, so the algorithm takes O(|V ||E|). Once network flow has been computed, the metric certifies each node for which there is flow from x− to the sink. Since any node with flow from x− to x+ also has flow from x− to the sink, any node through which trust flows is itself certified.”

There are two major problems with Advogato’s trust metric. First, the impact of a confused node “increases dramatically as it gets closer” to the seed. There is little reason to believe that a node 4 hops from the seed is four times harder to trick than a node 5 hops from the seed. The second problem exists even if we treat cx as the “cost” to corrupt a node. This problem arises because the attacker can increase a confused node’s cx during the attack. Recall that cx is assigned solely based on the distance from the seed to the

node. An example of this attack follows. In this example, nodes with some distance from the seed have capacity 400. Nodes with subsequent distances have capacities of 100, 25, 8, 2, and 1. Suppose an attacker confuses 400 nodes with cx = 1 and a single node y with cy = 400. This attack costs P x2C cx = 800. Based on the proof above, one might expect the attacker to be unable to get more than 399 of his nodes trusted. But if one attacker node certified by y certifies the rest of the confused nodes, those confused nodes all become cx = 25. The cost of the attack is only 800, but up to P x2C(cx −1) = 400 · 24+1 · 399 = 9999 bad nodes are accepted. [Jesse Ruderman, 2004]

From Jesse Ruderman's work [Jesse Ruderman, 2004], he showed that Advogato can be vunlernable to malicious attacks in a theoretic scenario.

The underlying code for this trust inference mechanism has been released under a free software license. Because of this, it has been the basis of numerous research papers on trust metrics and social networking including some researchers who argue that this metric can be attacked in certain scenario [Jesse Ruderman, 2004].

3.4 Appleseed

Appleseed [Cai-Nicolas Ziegler, Georg Lausen, 2004] is a group trust metric that uses spreading activation strategies.

function transform $(G = (V, E, C_V))$ {
  set $E' \leftarrow \emptyset$, $V' \leftarrow \emptyset$;
  for all $x \in V$ **do**
    add node x^+ to V';
    add node x^- to V';
    if $C_V(x) \geq 1$ **then**
      add edge (x^-, x^+) to E';
      set $C_{E'}(x^-, x^+) \leftarrow C_V(x) - 1$;
      for all $(x, y) \in E$ **do**
        add edge (x^+, y^-) to E';
        set $C_{E'}(x^+, y^-) \leftarrow \infty$;
      end do
      add edge $(x^-, \text{supersink})$ to E';
      set $C_{E'}(x^-, \text{supersink}) \leftarrow 1$;
    end if
  end do
  return $G' = (V', E', C_{E'})$;
}

Equation 2: Pseudo code for Appleseed

Marko Jung [Marko Jung, 2008] describes the Advogato trust algorithm as quoted below.

In 2004, Ziegler and Lausen (2004) developed Appleseed, a local group trust metric using concepts from spreading activation models in psychology to evaluate trust in Semantic Webs. One of Appleseed's biggest advantages is that it is scalable for huge network sizes as its performance rather depends on value of the energy injected into the system and the spreading factor than on the number of nodes in the network. There are also extensions available to

make the computation even faster: to hinder trust energy from overly covering vast parts of the entire network, the number of nodes which will be discovered can be limited. Another possibility to gain large speed-ups is to define an upper-bound for the path length (according to Milgram's (Milgram, 1970) 'six degrees of separation' paradigm, a maximum path length between three and six would be reasonable). Furthermore, Appleseed is highly resistant against attacks both through the introduction of a spreading factor, the normalisation of the trust statements and the fact that it satisfies the bottleneck property: nodes can not raise their impact by modifying the structure of trust statements they issue as the amount of trust accorded to an agent a only depends on its predecessors and does not increase when a adds more nodes. On the other hand, in addition to suffering from the same problems than most trust metrics, the Bootstrap problem and, in the case of local metrics, the introduction of a new user, finding appropriate values for Appleseed's parameters is rather difficult as spreading activation models are not commonly used in trust calculations and thus experience of the user will be needed. [Marko Jung, 2008]

This project is the first attempt to use a subjective user rating for estimating trust in software packaging. The Appleseed trust metric seems to be the best choice for this experiment because it is highly attack resistant and scales very well even in huge trust networks. In addition, Appleseed is highly customisable due to its numerous parameters and input variables. On the other hand, it is likely that many simulations will be needed to and the right parametrisation. [Marko Jung, 2008]

This mechanism is also attack resistant but requires performing normalisation on trust values estimation. A weakness in this algorithm is that in calculating this normalised trust value, it means a person who has made many high trust ratings will have lower value than if only one or two people had been rated. This is because trust is viewed as a limited resource in Appleseed and thus the total trust amount of trust is limited. This is not true in WBSN where it is possible to have very high trust for a large number of people and that trust should not be any weaker than the trust held by a person who only trusts smaller number of people.

Another weakness of this mechanism is that it requires exponentially higher computation with increasing number of users, thus it is not highly scalable due to this complexity.

3.5 SocialTrust

SocialTrust [James Caverlee, et al, 2008] is a reputation-based trust aggregation framework for supporting tamper-resilient trust establishment in WBSNs.

Two main features of this mechanism are its dynamic revision of trust by differentiates relationship quality from trust, and it includes a personalised user feedback mechanism to adapt to the social network as it evolves.

Initially all users are treated equally, then SocialTrust updates trust value through dynamic revision of trust ratings according to the following three components: the current quality component of trust, the history component, and the adaptation to change component, as illustrated in Equation 1. By

adding those three components, the The SocialTrust score "*ST(i,t)*"for a particular user i at time t can be evaluated.

$$ST(i,t) = \alpha \cdot Tr_q(i,t) + \beta \cdot \frac{1}{t}\int_0^t ST(i,x)dx + \gamma \cdot Tr'_q(i,t)$$

Equation 3: The Social-Trust algorithm score for user i at time t is defined as the above equation

By tuning α, β, and γ, the SocialTrust model can be optimized along a number of dimensions, e.g., (i) to emphasize the most recent behavior of a user in the network (via higher values of α); (ii) to de-emphasize the current user's behavior in the context of his entire history of behavior (via higher values of β); or (iii) to amplify sudden fluctuations in behavior (via higher values of γ). [James Caverlee, et al, 2008]

This mechanism is relatively new and has been tested on MySpace with over 5 million nodes and over 19 million relationship links. Its initial experiment results show that SocialTrust is more robust against malicious users where its trust inference value accuracy remains relatively unchanged as percentage of malicious users increases due to its link quality and feedback ratings mechanism.

SocialTrust compares its results with PageRank. PageRank is a rating mechanism used by Google search engine where high ranking is achieved if the sum of the ranks of its back links is high. It can be used to compare the effectiveness of a trust inference mechanism. However, having the content as popular does not necessarily indicate its accuracy or authenticity.

When a proportion of highly trusted users behave maliciously, PageRank has no mechanism for correcting this bad behavior. In contrast, the SocialTrust model incorporates link quality and feedback ratings into the trust assessment so that bad behavior is punished, and so the resulting precision measures are resilient to the presence of a large fraction of malicious users in the network. [James Caverlee, et al, 2008]

3.6 FuzzyTrust Algorithm

FuzzyTrust algorithm [Mohsen Lesani and Saeed Bagheri, 2006] considers the problem where trust inference in a large social network can encounter contradictory information. They propose fuzzy linguistic terms to specify trust to other users and developed an algorithm for inferring trust from a user to another user that is not directly connected of a WBSN.

Using fuzzy logic and its operators in trust models has been considered in many works. However, in the area of social networks, it is firstly proposed by Mohsen Lesani [Mohsen Taherian, et al, 2008].

The pseudo code of the FuzzyTrust algorithm is quoted below

Forward wave: The required strength fuzzy set computation

- Iterate the nodes from the source to the sink similar to the breadth first search level by level to find shortest paths.
- Set the path strength fuzzy set from source to any node in the next level using the previously set path strengths fuzzy sets from the source to current level nodes.

Backward wave: The trust from nodes to the sink inference

Equation 4: SThe FuzzyTrust algorithm pseudo code [Mohsen Lesani and Saeed Bagheri, 2006]

This algorithm computes trust from stronger and shorter paths as it performs a breadth-first-like search through the nodes to find shorter paths and also to find the path from source to sink strength fuzzy set. This is a similar approach to TidelTrust algorithm [Jennifer Golbeck, 2005] and as a consequence, FuzzyTrust suffers the same weakness as the TidelTrust algorithm where certain information is discarded as only the nodes on the shorter paths are considered, as well as the whole path is not considered in the shorter path.

FuzzyTrust algorithm is designed to handle conflicting trust values by using fuzzy linguistic expression (e.g. low, medium, high), which is easier for users to assign trust because those linguistic expression are natural to people than to use numerical expression (e.g. 1, 2, 3, …, 9, 10).

This algorithm was tested by simulating and comparing with TidelTrust algorithm [Jennifer Golbeck, 2005] but the results of Lesani's simulation indicate that the fuzzy algorithm offers more accurate information than the TidalTrust and it provides more meaningful information when there is conflict of information from different sources of information whereas TidalTrust would have average the values of the conflicting sources of information and this conflict information is lost in this averaging.

In another word, the TidalTrust algorithm simply takes the average of 0 and 10, yielding the 5 value that is a value just in between (medium), none of the neighbors believes in it in fact. The FuzzyTrust algorithm with Larsen method result fuzzy set is the fuzzy set equal to "Low or high" linguistic expression fuzzy set. While FuzzyTrust result is exactly the information that exists in the trust graph, TidalTrust algorithm fails to exactly convey the information existing in the trust graph due to its improper averaging scheme that it employs for composition. Similar arguments can be brought for other node pairs that two algorithm results do not completely agree. [Mohsen Lesani and Saeed Bagheri, 2006]

3.7 SUNNY

SUNNY [Ugur Kuter and Jennifer Golbeck, 2007] is a trust inference algorithm that uses a probabilistic sampling technique to estimate a user's confidence in the trust information from designated sources. It computes an estimate of trust based on only those information sources with high confidence estimates, regardless path length, to achieve higher accuracy trust estimates. This is in contrast to TidalTrust where only shortest paths are considered.

It differs to other algorithm in that it claims to be the first trust inference algorithm which includes a confidence measure in its computation. It is more accurate against Golbeck's own TidalTrust when tested with data from FilmTrust social network.

Procedure SUNNY(T, n_0, n_∞)
$B_T \leftarrow$ GENERATEBN(T)
for every leaf node n in B_T do
$decision[n] \leftarrow$ UNKNOWN
$\langle P_\perp(n_0), P_\top(n_0)\rangle \leftarrow$ SAMPLE-BOUNDS(B_T)
for every leaf node n in B_T do
set the lower and upper probability bounds such that
$P_\perp(n) = P_\top(n) = 1.0$
$\langle P'_\perp(n_0), P'_\top(n_0)\rangle \leftarrow$ SAMPLE-BOUNDS(B_T)
if $|P'_\top(n_0) - P_\top(n_0)| < \epsilon$ and $|P'_\perp(n_0) - P_\perp(n_0)| < \epsilon$
then $decision[n] \leftarrow$ TRUE
else $decision[n] \leftarrow$ FALSE
return (COMPUTE-TRUST$(B_T, decision)$)

Equation 5: SUNNY, the procedure for computing trust and confidence in a trust network.

Ugur Kuter *et al* [Ugur Kuter and Jennifer Golbeck, 2007] describe SUNNY trust algorithm as quoted below.

SUNNY is a trust inference algorithm that uses a probabilistic sampling technique to estimate our confidence in the trust information from some designated sources. SUNNY computes an estimate of trust based on only those information sources with high confidence estimates. In our experiments, SUNNY produced more accurate trust estimates than the well known trust inference algorithm TIDALTRUST (Golbeck 2005), demonstrating its effectiveness [Ugur Kuter and Jennifer Golbeck, 2007].

SUNNY clams to be the first trust inference algorithm that includes a confidence measure in its computation. SUNNY performs a probabilistic logic sampling procedure as in (Kuter et al. 2004) over the Bayesian Network generated by our representation mapping. In doing so, it computes estimates of the lower and upper bounds on the confidence values, which are then used as heuristics to generate the most accurate estimates of trust values of the nodes of the Bayesian Network [Ugur Kuter and Jennifer Golbeck, 2007].

The experimental evaluation of SUNNY in comparison to the well-known existing work on TidalTrust (Golbeck 2005) shows that SUNNY significantly outperforms TIDALTRUST [Ugur Kuter and Jennifer Golbeck, 2007].

3.8 Trusted Gossip

Trusted Gossip was proposed by the two scholars [Arindam Mitra and Mucumaru Maheswaran, 2007] where it is a trusted gossip protocol for disseminating popular information from reputable users while restricting flow of spam messages. This is so that good information can still be spread effectively by word of months while reduces the negative effect of unwanted information that we call spam.

The algorithm estimates trust via a Bayesian trust estimation process where information are tagged and processed, then the trust value is evaluated through a recommender system.

They present three approaches for this recommender system. Receiver Initiated, Sender Initiated, or using both as Hybrid, where node-level filtering and message-level filtering is implemented at the sender and/or the receiver side.

Their research uses a subset of Flickr social network users and find this subset of randomly selected users can reach most other users within 6 hops, but no path is found between many users, where inference trust can not be evaluated. They also find 3 hops is the limit for likeminded nodes where trust evaluation is more accurate.

Their main findings are that message level filtering done at the originating nodes is the more accurate method of determining inferred trust.

Furthermore, they perform experiment their trust mechanism in three WBSNs (Flickr, Bookcrossing, and Movielens) where they prove that this Trusted Gossip mechanism is robust against reputation distribution. This means regardless of whether there is larger number of highly reputed users or larger number of lowly reputed users in a WBSN, their trust inference estimation has negligible error differences.

3.9 RN-Trust

In this algorithm [Mohsen Taherian *et al*, 2008], it adopts a resistive network into a trust inference network. The concept is similar to the idea of electrical resistance, where a resistive circuit network is a collection of resistors that represent trust value as reverse of the resistance.

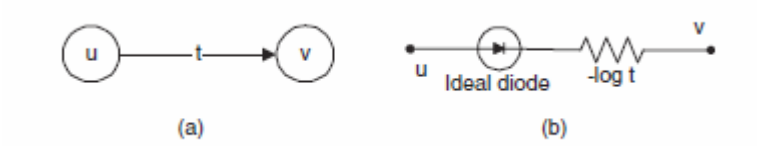

Figure 7: (a) An edge in a trust network with the trust value t from u to v; (b) Corresponding edge in the resistive network.

To calculate the inferred trust value between the source nodes and the target node t, first the equivalent resistance between s and t is computed. Suppose that the equivalent resistance between s and t is $R_{eq}(s, t)$. The inferred trust value between s and t, called T(s, t), can be calculated from the following equation. [Mohsen Taherian, et al, 2008]

$$T(s,t) = 10^{-R_{eq}(s,t)}$$

Figure 8: RN-Trust algorithm

The principle of how RN-Trust algorithm works is best described by Mohesen Taherian [Mohsen Taherian, et al, 2008] in their paper as quoted below.

In addition to being simple, the algorithm resolves many problems of the previous ones. Basically, the main idea is to use the Resistive Network concept to simulate trust networks. A resistive network is a collection of

resistors which are connected in series and parallel. In the proposed model, each node in the trust network is mapped to a node in the resistive network. Also, for each two adjacent nodes in the trust network, a resistor is placed between their corresponding nodes in the resistive network. It is clear that the resistors' values must have a reverse relation with the trust values. After constructing this resistive network between the source and the sink, the equivalent resistance of this electric circuit can be used as a measure to calculate the trust value from the source to the sink. In RN-Trust, there is no need to ignore any information. The problem of inferring trust through a chain is resolved too. In addition, the algorithm is very simple and its time complexity is polynomial. In fact, it requires $O(V^3)$ time in the worst case in which V is the number of nodes in the trust network. In this algorithm, the trust network is mapped to a resistive network, and a voltage source connects from u to v, while the electrical current flows between u and v. This current can be interpreted as the trust relation from u to v. If there is a resistor between u and v, then the amount of current flows from u to v decreases. Thus, the higher the trust value, the lower the value of the resistor [Mohsen Taherian, et al, 2008].

RN-Trust is a trust network is modeled with a resistive circuit. That is, each trust relationship is mapped to a resistor beside an ideal diode. The role of diode is satisfying the asymmetric property of trust in the real world. A logarithmic function is used to map trust values to resistors' values. Furthermore, some required properties of the model were introduced and it was investigated how RN-Trust satisfies them. We discussed how trust inference with RN-Trust takes out some major problems of previous algorithms. Also, the time complexity of RN-Trust algorithm is calculated and proved to be a polynomial time which has not significant difference with previous algorithms [Mohsen Taherian, et al, 2008].

This mechanism has enough generality to being applied in other environments such as peer-to-peer networks and multi-agent systems [Mohsen Taherian, et al, 2008].

Similar to how TidalTrust [Jennifer Golbeck., 2005] assigns trust values, the trust values in this algorithm are continuous values in the range of [0, 1].

The RN-Trust algorithm resolves many problems of other trust inference algorithms, especially in regard to TidalTrust [Jennifer Golbeck., 2005]. For instance, it does not need to ignore trust inference derive from long path by consider the full length of the trust inference path, rather than only the shortest ones, giving a more complete overview in determining the inference trust value. This addresses the major weakness in the TidalTrust algorithm [Jennifer Golbeck., 2005] where some information along the path which may affect the accuracy of the calculation is ignored. In addition, the algorithm is very simple and its time complexity is polynomial, thus highly scalable.

4 Findings and Discussion

In this section, we discuss our findings on how scholars have different methods of developing their trust inference mechanism in sub section 4.1. Then in sub section 4.2, we present our analysis on the merits and weaknesses of those trust inference mechanism in solving the problem of accurately determining the trust inference in a WBSN. In addition, possible issues for future studies are presented in sub section 4.3.

4.1 Approach of the Problem

Scholars approach the issue of developing a trust inference mechanism from different perspectives to solve the problem of estimating reliable trust inference value while being robust against malicious attacks from users providing false trust values.

Firstly, algorithms such as TidalTrust [Jennifer Golbeck., 2005], Binary Trust [Jennifer Golbeck, and James Hendler, 2006], and FuzzyTrust [Mohsen Lesani and Saeed Bagheri, 2006] make the assumption that only trust values in the shortest or strongest paths are included in their calculation to achieve simplicity with sufficient accuracy.

Secondly, Advogato trust metric [Raph Levien, 2002] uses a network flow model and certification process to control how trust is inferred.

Thirdly, a trust inference mechanism can perform more than one step of trust estimation. For example, Trusted Gossip algorithm [Arindam Mitra and Mucumaru Maheswaran, 2007] performs a Bayesian trust estimation process, and then performs a recommender system, to achieve higher accuracy and improve robustness against reputation distribution.

Fourthly, SUNNY [Ugur Kuter and Jennifer Golbeck, 2007] includes the confidence factor and [Mohsen Taherian *et al*, 2008] uses an electrical resistance model to eliminate the weakness of only considering the shortest or strongest paths in trust inference value estimiation, such as in TidalTrust [Jennifer Golbeck., 2005] where only trust values in the shortest paths are considered.

Lastly, there are algorithms which attempt to solve different aspect of trust scenarios. For instance, FuzzyTrust [Mohsen Lesani and Saeed Bagheri, 2006] attempts to resolve contradictory information by using fuzzy linguistic terms instead of numeric trust values.

4.2 Merits and Weaknesses

All trust inference mechanisms described in this paper are more accurate in estimating trust inference values than taking simple average along the path of the trust inference. Mechanisms such as Trusted Gossip researchers [Arindam Mitra and Mucumaru Maheswaran, 2007] show they are robust against variation in trust value distribution where their accuracy is relatively unaffected by what trust values are in the WBSN. Other mechanisms such as Binary Trust [Jennifer Golbeck, and James Hendler, 2006] algorithms and SocialTrust [James Caverlee, et al, 2008] are designed to be relatively unaffected by malicious attacks while scalable in their application in large size WBSN with over a million users.

One weakness that applies to all of the trust inference mechanism described in this paper is that none has comprehensive tests on various

WBSN to evaluate their suitability with certain types of WBSN. Furthermore, they are not fully independently tested by another research organisation or commercial entity to verify their experimental calculations. This is an indication that the topic is still relatively new and the implementation of those mechanisms in software applications has not been widely accepted.

Other common weaknesses that apply to some of the trust inference mechanisms are described below.

Firstly, complexities of some mechanisms such as in the Appleseed [Cai-Nicolas Ziegler, Georg Lausen, 2004] and Trusted Gossip algorithms [Arindam Mitra and Mucumaru Maheswaran, 2007] show they may have scalability issues. WBSN that has over millions of members may prove prohibitive for these mechanisms. This is why some mechanisms aim to have efficiency as one of their objectives such as the RN-Trust [Mohsen Taherian *et al*, 2008].

Secondly, relationships and trust are dynamic. New relationships are formed everyday and old ones weaken gradually. Some mechanisms take this into account and include in its mechanism a method of updating its trust rating as illustrated in SocialTrust [James Caverlee, et al, 2008].

Thirdly, existence trust inference mechanisms can not adjust to various real life scenarios to give consistently accurate trust inference value while this adjustment property is needed in WBSNs to reflect how people evaluate trust. For example, the algorithm used in a WBSN with generic topic and high number of malicious users may not be suitable to estimate trust inference value in a WBSN with specific topic and high level of confidence in the performance of most users in that WBSN.

A possible general solution to this weakness is to develop a grand unified trust inference mechanism that can adjust its algorithm and logics depending on the topic to provide the most consistently accurate trust inference value estimation. Additionally, this mechanism needs be applicable to not only in WBSN, but also in P2P networks.

4.3 Possible future studies

From the analysis above, this paper proposes 5 topics for possible future studies in the area of trust inference mechanism. They are designed to improve current trust inference mechanisms and/or to introduce scenarios where a trust inference mechanism can be applied outside existing known applications.

1. Further Improve upon the accuracy and the precision (consistently accurate in different trust inference scenario or WBSN) of trust inference value estimation. This is because none of the trust inference mechanism can estimate trust inference value close to 100% accuracy or attending 95% accuracy which is generally a typical WBSN user would expect if he were to make any reliable decision upon this inference trust estimated by the mechanism.
2. Develop an event based trust inference algorithm where only past events are used in the calculation so that it is context focus and can continuously update its trust inference value. Intuitively, human being earns his trust with another human being from prior experiences of interacting with each other. A trust inference algorithm that purely based its calculation on past

events or heavily weighted on past event would simulate how people perceive and evaluate trust. For example, Alan trusts Billy in his opinion in music, and Billy trusts Clark in his opinion in technology. Alan wants to buy a wireless modem but Billy never owns one, so Alan seeks the opinion of Clark via Billy and Clark suggested Billy to tell Alan to buy a particular brand and model. One week later after Alan buys that model, he finds that this wireless modem is not what he expected to be, thus he tells Billy about this. Now both Billy's trust to Clark may or may not change, but Alan's trust inference value should greatly decrease due to this experience. A past event based trust inference model can be developed to include this scenario and reflect more accurately how people view trust. This method also considers the dynamic nature of trust where trust value may not necessarily stays constant over time and often changes dramatically after an event between two people has occurred.

3. Develop context aware extensions so that the WBSN may support multiple trust views of each user depending on the context or topic of discussion to improve accuracy and relevance of the trust inference. For example, Alan may trust his grandfather David's opinion on antiques and vintage cars, but not so for computers and new sports cars. A new trust inference mechanism may categorises different context and assign different trust value depending on that context.
4. Pushing the boundaries of how trust inferences can be applied in WBSN and be able to utilise it more than a simple email filtering system as illstrated in TrustMail using Binary Trust [Jennifer Golbeck, and James Hendler, 2006]. A suggested application can be in the area of Artificial Intelligence; in particularly, the use in military intelligence gathering where a government needs to determine the trustworthiness of intelligence reports from its agents. Furthermore, if a trust inference algorithm can be applied in a commercial WBSN or application, similar to how Best Path Selection algorithm can be applied to shipping and transport companies, then there will be accelerated growth in the development of trust inference algorithm, possibly in the development of an open standard and/or a proprietary algorithm.
5. Studies on the requirement of computing resources such as CPU processing power, memory usage, storage requirement, and network traffic generated, when a trust inferring mechanism is applied in a WBSN. This will provide a cost estimate if a commercial entity wishes to adopt a particular trust inference mechanism in its WBSN. In turn this will also encourage the use of trust inference mechanism in general. For example, a company may have its own intranet with a social network of its employees. A particular employee may seek information outside his direct contacts (e.g. he seeks someone with a certain programming language development experience in building a particular application module), then a trust inference mechanism can automatically provide him with a list of suggested employees he can contact with.

5 Conclusion

In this paper, we define what trust inference is in a WBSN. We then review a selection of research papers to examine existing trust mechanisms, their properties, and their approach to trust inference calculation.

We discuss some merits and weaknesses in those trust inference mechanisms, considering factors such as accuracy, scalability, robust against reputation distribution, and attack resistance from malicious users.

From the result of this paper, we can surmise that the maturity of a generally accepted trust inference mechanism is yet to be realised.

Further more; we propose 5 possible future research works which are either in area that is yet to be explored by existing mechanisms, or in area which existing mechanisms can be improved.

References

Arindam Mitra, Muthucumaru Maheswaran, [2007] *Trusted Gossip:A Rumor Resistant Dissemination Mechanism for Peer-to-Peer Information Sharing.* Advanced Information Networking and Applications, 2007. AINA '07. 21st International Conference on. Dept. of Comput. Sci., Manitoba Univ., Winnipeg, MB.

Cai-Nicolas Ziegler, Georg Lausen [2004] *Spreading Activation Models for Trust Propagation.* Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service (EEE'04)

Cambridge Dictionaries Online [2009] *Definition Trust,* viewed on 22 Mar 2009. http://dictionary.cambridge.org/define.asp?key=85211&dict=CALD

James Caverlee, Ling Liu, Steve Webb [2008], *Towards Robust Trust Establishment in Web-Based Social Networks with SocialTrust.* Proceeding of the 17th international conference on World Wide Web. Beijing, China

Jennifer Golbeck., [2005] *Computing and Applying Trust in Web-based Social Networks,* Doctor of Philosophy Dissertation, University of Maryland, College Park.

Jennifer Golbeck, James Hendler [Nov 2006] *Inferring Binary Trust Relationships in Web-Based Social Networks.* ACM Transactions on Internet Technology, Volume 6, Issue 4, New York, NY, USA.

Jesse Ruderman [2004] *A comparison of two trust metrics*. UCSD CSE

Marko Jung [2008] *Appleseed as Trust Metric for the openSUSE Build Service. Bachelor Seminar Telecommunications Lab, Universit□at des Saarlandes Prof. Dr.-Ing. Thorsten Herfet*

Mohsen Lesani and Saeed Bagheri [2006] *Fuzzy Trust Inference in Trust Graphs and its Application in Semantic Web Social Networks.* World Automation Congress, 2006. WAC '06. Sharif University of Technology, Iran.

Mohsen Taherian, Morteza Amini, Rasool Jalili, [2008] *Trust Inference in Web-Based Social Networks Using Resistive Networks.* Internet and Web Applications and Services, 2008. ICIW '08. Third International Conference on 8-13 June 2008. Page(s): 233-238.

Piotr Sztompka [1999] *Trust: A Sociological Theory.* Cambridge University Press, Cambridge, UK.

R. Guha, Ravl Kumar, Prabhakar Raghavan, Andrew Tomkins [2004] *Propagation of Trust and Distrust.* Proceedings of the 13th international conference on World Wide Web, New York, NY, USA

Raphael L. Levien [2002] *Attack resistant trust metrics. PhD thesis,* Department of Computer Science, University of California, Berkeley.

Stephen Paul Marsh [Apr 1994] *Formalising trust as a computational concept.* PhD thesis, Department of Mathematics and Computer Science, University of Stirling.

Ugur Kuter and Jennifer Golbeck [2007]. *SUNNY: A New Algorithm for Trust Inference in Social Networks, using Probabilistic Confidence Models.* Proceedings of the Twenty-Second National Conference on Artificial Intelligence (AAAI-07). Vancouver, British Columbia, July, 2007.

Webster's Online Dictionary [2009] *Definition: Trust*, viewed on 22 Mar 2009. http://www.websters-online-dictionary.org/definition/trust

Yarden Katz, Jennifer Golbeck [2007]. *Using Social Network-based Trust For Default Reasoning On The Web.* Submitted to Journal of Web Semantics, 2007.

Young Ae Kim, Minh-Tam Le, Hady W Lauw, Ee-Peng Lim, Haifeng Liu, Jaideep Srivastava,. [Apr 2008] *Building a web of trust without explicit trust ratings* .Data Engineering Workshop, 2008. ICDEW 2008. IEEE 24th International Conference. Page(s): 531-536.