

# Trust in Opportunistic Networks

Sacha Trifunovic, Franck Legendre  
Computer Engineering and Networks Laboratory  
ETH Zurich, Switzerland  
{lastname}@tik.ee.ethz.ch  
TIK Report 318

## ABSTRACT

Opportunistic networks enables mobile users to participate in social interactions through applications such as content distribution, flea-market, micro-blogs and round based games. To interact securely, the establishment of trust is vital in a distributed environment. Trust is required to validate an identity and avoid sybil users, select trustworthy interaction partners or to collect useful opinions in a recommender system. Different forms of trust, either based on a social connection (friend), frequent encounter (familiar) or similar interests, can be harnessed in order to best suit the different requirements. Algorithms are proposed in order to evaluate different forms of trust in a distributed manner and combine them for different requirements. Complexity, trust propagation characteristics and security issues are discussed and thoroughly analyzed using synthetic models and real world mobility traces.

## 1. INTRODUCTION

With the advance of mobile devices, a variety of new networking scenarios and applications are emerging. As humans take advantage of their mobility and interact and cooperate with other humans, so can mobile devices, giving birth to opportunistic networking. Users would then be involved in participatory social interactions with their surrounding using applications such as mobile social networking [1], content distribution [2], flea-markets, micro-blogs [3], and round-based games. Most of these applications will rely on the publish/subscribe paradigm [4] where users will publish their inputs or services (e.g., content, sold objects, blog entries, game opponents) and subscribe based on their solicitations (user-driven, one-hop downloads) avoiding routing per se. Inputs will spread from their authors (or contributors) to consumers through relays (or suppliers) in a delay-tolerant epidemic fashion from hop to hop using mobility. Possible areas of operations range from remote and rural areas, occasional events (e.g., conferences, exposition halls) to stationary and settled communities (e.g., work teams, military bases), and even city-wide scale areas.

In such open wireless networking environments, one of the key challenges is security. The cornerstone of every security system that involves multiple users is the establishment of trust. In a decentralized and distributed environment, where no central infrastructure can be assumed, this is especially challenging. Trust is required for various tasks in mobile environments.

In a distributed identity management system, only the inimitability of an identity but no one-to-one binding between entity and identity can be achieved by cryptographic means. Trust in an identity is required in order to verify its legitimacy and to avoid sybil attacks. Most applications do not necessarily need a one-to-one binding between an identity and a user but rather to establish trusted relationships between the interacting agents such as service providers and service consumers (and avoid impersonation once these trusted relationships are established). Trust in the service or interaction partner is required in order to avoid fake IDs and promote the interaction with legitimate users, thus improving the user experience. Whenever an additional reputation or recommender system is present, that takes social reputation into account [5], trust in the truthfulness of a rating is also required in order to form a factual opinion.

Trust comes in various forms. We trust our high school friend because we have spent a lot of time with her, know her surrounding and family as well as her character and habits and thus, are able to assess what she is capable of. We also trust a fellow student we do not know but see in most of the classes. To a certain degree this is because we know his main occupation, but more importantly, we can expect to see him in the same classes next week. Furthermore, we even trust a film critic, whom we have never met in our lives, but who has given advice that mirrored our taste. We trust the critic to provide more useful advice to us in the future.

In this report we try to harness different forms of trust and apply them to the different requirements. Firstly, we use *social trust*, established through consciously defined friend ties, making use of the fact that mobility helps peer-to-peer security [6]. Secondly, *environmental trust* is inferred from the familiarity of the surrounding peers leveraging the complex network structure [7] resulting from the social nature of opportunistic networks. Thirdly, *similarity trust* is assessed based on the correlation of the ratings or opinions among different users similar to [8, 9]. Obviously trust can also be inferred of the direct ratings of past interactions but we leave this topic to the scope of reputation systems. All these different forms of trust are orthogonal to each other and can be assessed independently. Furthermore, these forms of trust can be weighted and combined according to the different requirements. For the identity management system, social trust might be the most important one, but in order to assess a correct rating out of various recommendations, the similarity trust seems more fit.

The rest of the report is organized as follows: the next section presents some related work, followed by the definition and comparison of reputation and trust. Section 4 explains the establishment of the different forms of trust. The algorithms presented in this section are evaluated in Section 5. Section 6 concludes this report.

## 2. RELATED WORK

In classical networks, trust is established by a certificate authority (CA) through a public key infrastructure (PKI) [10]. In a distributed environment this approach becomes useless, since no fixed infrastructure and thus no authorities are available. In an opportunistic or ad hoc network, the CA duty can be handed over to nodes which then have to generate their own credentials and sign certificates of others once secure pairing is performed. Making use of the mobility of the nodes, trust can be established between friends [6] or in small groups [11] using a secure pairing protocol. Both approaches have the drawbacks of limiting trust to only a small amount of users and requiring an conscious interaction in order to establish trust. A more flexible approach is described in [12], where certificate chains, similar to PGP [13], are built. Unfortunately, this approach requires routing, in case missing links in the chain have to be filled, which is unavailable in opportunistic networks. Especially for complete strangers and new users this is an additional problem. Furthermore, its arguable whether transitive trust over several hops should be as strong as in a direct friend.

The social analysis of graphs have inspired a rich set of complex social network tools. Links can be predicted by analyzing different measures of similarity [14] which can be used for routing in delay tolerant networks [15]. As a more advanced tool, communities may be detected by optimizing modularity greedily [16] or if speed is an issue, hierarchically [17]. This can also be done in wireless networks [18] since the node's mobility reflects the underlying social structure. Identifying communities also helps routing in such environments [19]. In mobile networks special attention has to be paid to the contact aggregation in the graph [20] and aging [21] in order for this tools to remain effective. Especially the aging issue lacks a proper solution. To our knowledge, never have any of these tools been used to infer any kind of trust where measures like familiarity<sup>1</sup> and similarity place a predominant role.

Many reputation and recommender systems have been proposed in the past [22, 23] which take individual as well as social reputation [5] into account. Although some use a deviation test in order to filter out liars, only few introduce a measure of trust in order to make better use of social reputation. Liu and Issarny [8] evaluate the similarity of direct ratings and recommendations to assess trust in future opinions. Walsh and Sierer [9] take it one step further and propose Credence, a reputation system for peer-to-peer filesharing that allows the correlation of ratings over various hops in order to achieve trust in certain content. Other systems ensure trust through a gossiping protocol using witnesses [24, 25] but when trust has to be assessed of over various hops the selection of witnesses is hardly feasible.

---

<sup>1</sup>Accumulated time or frequency of contacts.

The main reason why trust is required in such a distributed setting, is the simplicity of creating and leaving multiple identities and launching sybil attacks [26]. By these means, an attacker might gain a larger influence, abandon bad reputation or evade responsibility of his/her actions. Various methods have been proposed in order to prevent such attacks. Piro et al. [27] argue that sybil users can only communicate serially and would thus cause much fewer collisions on the MAC layer and are thus detectable. SybilGuard [28] approaches this problem from a different angle by assuming that sybil users can create many identities but only few trust relationships and can thus be detected by carefully observing the social graph. Location based sybil detection is also an effective measure [29] but requires specialized hardware. However, the presented approaches do not prevent users explicitly from generating multiple identities and only provide a probabilistic assessment of a node using Sybil identities. But all approaches are completely independent of each other and orthogonal to the establishment of trust. They can thus be used as an additional measure to increase the complexity of an attack and the probability of detection.

### 3. TRUST VS. REPUTATION

Trust is often confused with reputation and used interchangeably. To avoid that, we give clear definitions of both terms and discuss their relation to each other. *Trust is a particular level of the subjective probability with which an agent assesses that another agent or group will perform a particular action, both before he can monitor such action (or independently of his capacity ever to be able to monitor it) and in a context in which it affects his own action* [30]. In contrast to trust, which tries to predict a future action, reputation is a passive property depending on past actions: *Reputation of an agent is a perception regarding its behavior norms, which is held by other agents, based on experiences and observation of its past actions* [8]. Of course there is a strong relation between both terms, since an user's good reputation promotes the trust in that user. Nevertheless, in this paper we ignore the concept of reputation and the trust that might be assessed based on it and focus only on other ways of establishing trust. The reason for that is that either the proposed trust metrics should be usable as an input for the reputation system in the first place or be used totally independent from it, for example to counteract sybil attacks.

### 4. TRUST METRICS

In this paper we propose three ways of establishing trust: through social connections, based on the environment, based on similarity/taste. A fourth way would be to base trust on the experience of direct interaction, but this belongs to the scope of reputation systems. All four methods are orthogonal to each other and can be used in parallel. The different trust values can be combined in order to cope with the advantages and disadvantages of each method.

#### *Social Trust.*

The cornerstone of social trust are consciously selected friend ties. Taking advantage of the mobility of the devices, secure and reliable friend ties can easily be achieved via any form of secure pairing. Similar to PGP [13] and Capkun et al. [12] we assume some transitivity of trust. According to Swamyathan et al. [31] this is a reasonable assumption over up to 6 hops. Taking conscious friend connections as a basis of trust has its advantages as well as disadvantages. Through the secure pairing process, the human entity behind the identity is verified and makes sure it is not a sybil ID. Secondly, the trust is based on a social relation, which assumes a certain knowledge about the persons personality and what he/she is likely to do and what not. On the downside, it requires the interaction of the user. Furthermore, the resulting graph of friend ties is loosely connected and regular interactions are not guaranteed. It does not guarantee to give us a trust assessment about a lot of nodes that are met regularly.

#### *Environmental Trust.*

In everyday life, there are certain individuals we regularly share the same space or the same activity with. For example, there are usually the same people in our apartment buildings, mostly the same coworker at the office every day, and some people that go regularly to the same gym each Tuesday and Thursday. A community detection algorithm can try to identify these communities by carefully observing the environment and analyzing the peers in the area over time. Although communities have a social notion, this is not guaranteed, since members of a detected community doesn't have to know or be friends with each other, but only share the same environment for a significant amount of time.

Community detection cannot guarantee for a certain entity to be behind the proclaimed identity, thus this method is not as secure as basing the trust on conscious friend ties. Nevertheless, a certain amount of trust in a familiar stranger can be justified since the identity cannot be a fast living, which is useful against sybil attacks. This increases the effort of an attack significantly. A significant advantage this method has over the friend ties, is that it requires no interaction of the user. An important aspect to keep in mind is that current community detection mechanisms are easily tampered with or at least influenced so special care has to be taken in making it resilient to attacks.

#### *Similarity Trust.*

Certain trust can also be based on similarity/taste by correlating ratings or opinions of different users. This method is usually combined with or part of a reputation system. Such a system for mobile ad hoc networks is introduced by Liu and Issarny [8]. There, trust is assessed by comparing recommendations of other users with direct experiences. Additionally, Walsh and Sireer [9] propose Credence, a reputation system for peer-to-peer filesharing that allows the correlation of ratings over various hops in order to achieve trust in certain content. Although this method might identify unwanted content, or at least select peers, which for their similar taste, are more relevant in the given context, it has significant weaknesses. Firstly, it requires the presence of a reputation system in the first place and the user needs to have experience/ratings already. Secondly and more importantly, recommendation might easily be forged and manipulated to be similar with the respective peer in a decentralized environment. Making this way of assessing trust secure, especially for opportunistic networks, is not an easy task which is beyond the scope of this report and is left for future work.

### **4.1 Social Trust**

Social trust requires a user to consciously create friend ties. Through a secure pairing process the nodes can sign each others certificates as proof of their friendship. We do assume some transitivity of trust as mentioned before, but instead of just building chains as in PGP [13], we take into consideration that trust in a user we have several common friends with is higher than in a user we have a single connection over various hops.

Nodes that meet in an opportunistic way exchange their list of friends and build up a friendship graph  $G_F$ . The friendship graph is a special graph organized in levels  $L_d$  constituted by a set of nodes having the same distance  $d$  from the local node  $n_0$ . Edges only exist between nodes in sequenced levels. The graph can be constructed using a slightly altered breadth-first search (BFS) algorithm. The modification is necessary to allow various edges from nodes in  $L_d$  to end in a single node in  $L_{d+1}$ . A trust value  $t_i$  in the range  $[0, 1]$  for every node in the friendship graph  $G_F$  is then calculated by Algorithm 1.

The algorithm gives all direct friends a trust value of 1. The values become smaller with each hop distance, depending on how well a node is connected. The values decrease faster if many friend ties exist, and does not decrease at all if only one node exists on each distance level. For this reason the minimum degradation factor  $c$  is introduced. It is recommended to set  $c$  to a value of 2 or 3, to force a certain trust degradation per distance in a sparse friendship graph.

An thorough evaluation of the performance, the expected trust distribution, and of security concerns will be given in Section 5.1.

### **4.2 Environmental Trust**

As argued before, the careful observation of ones surrounding can increase trust in certain nodes. One way of analyzing the surrounding is by community detection. A community can be described as a well linked clustering of entities [21]. By analyzing contact duration and/or contact frequency of the surrounding peers and sharing this information with those, one can try to guess the communities to which one belongs. In [18], three algorithms *Simple*, *k-Clique*, and *Modularity* are proposed. Each algorithm needs to maintain and exchange at least the set of familiars and the community members. Users are added to the familiar set after their familiarity (accumulated time or frequency of being in proximity) surpassed a certain value. In order to build up the community each algorithm uses a different strategy. The *Simple* algorithm compares the similarity of the surrounding of two nodes in order to decide whether to add each other into their respective community. The *k-Clique* algorithm searches for cliques of a certain size a node belongs to and the *Modularity* algorithm tries to find a community by optimizing its modularity as in most state of the art community detection algorithms [16, 32, 17].

---

**Algorithm 1** Social Trust

---

$n_i$ : A node (local node:  $n_0$ )  
 $e_{i,j}$ : Edge from  $n_i$  to  $n_j$   
 $FR_i$ : Set containing all friends of  $n_i$   
 $G_F$ : Friendship Graph of  $n_0$   
 $ts_i$ : Social trust value of  $n_i$   
 $L_d$ : Set of nodes with distance  $d$  from  $n_0$  in  $G_F$   
 $c$ : Minimum degradation constant  
 $L_0 = n_0$   
 $ts_0 = 1$   
**for all** nodes  $n_i$  in proximity **do**  
  acquire  $FR_i$  from  $n_i$  and update  $G_F$   
  build  $G_F$  and get  $L_d \forall d$   
  **for all**  $d > 0$  **do**  
    **for all**  $n_j$  in  $L_{d+1}$  **do**  
      
$$ts_j = \frac{1}{\max(|L_d|, c^d)} \cdot \sum_{n_k \in L_d: \exists e_{k,j}} ts_k$$
  
    **end for**  
  **end for**  
**end for**

---

For the assessment of trust, finding a community with an optimal modularity is not really helpful, since the modularity does not say anything about the trustability of nodes. More important is the familiarity and the similarity of the nodes. The more familiar a node is, the longer it has been in proximity, which means the probability of it being a short living user is smaller and the effort to create influential sybil users is bigger. The more similar a node is, the more nodes from the surrounding share this opinion of the node being familiar.

---

**Algorithm 2** Environmental Trust

---

$n_i$ : A node (local node:  $n_0$ )  
 $f_{i,j}$ : Familiarity value  $n_i$  has for  $n_j$   
 $F_i$ : Set containing  $f_{i,j}$  of all  $n_j$   
 $te_i$ : Environmental trust value of  $n_i$   
 $fs_i = \sum_j f_{i,j}$   
**for all** nodes  $n_i$  in proximity **do**  
  update  $f_{0,i}$   
  acquire  $F_i$  from  $n_i$   
  **for all**  $n_j$  **do**  
    
$$te_j = \frac{f_{0,j}}{fs_0} + \sum_k \frac{f_{0,k}}{fs_0} \cdot \frac{f_{k,j}}{fs_k - f_{k,0}}$$
  
  **end for**  
**end for**

---

Algorithm 2 calculates a trust value in the range  $[0, 2)$  based on the familiarity and similarity of the nodes, although values greater than 1 are negligibly rare. It requires each node to keep track of the connection times with the surrounding nodes, in order to keep the familiarity values  $f_{0,i}$  up to date. The set of all familiarity values is exchanged with all neighbors thus giving a node a local approximation of the weighted network graph. The algorithm then assigns trust values to nodes up to 2 hops apart from the local node depending on their familiarity and similarity.

This algorithm has similar weaknesses as any distributed community detection algorithms we could find. They are the normalization of familiarity values so they can be compared between nodes and a proper aging of the familiarity values. Hossmann et al. [20] solve a similar contact aggregation problem by finding the optimal edge density of the graph. They propose a online optimal density tracking algorithm that infers the desired density by optimizing the community structure of the graph. The same principle

can be used to dynamically find a suitable aging speed that produces the optimal density of the graph and comparable familiarity values. This can be done for example by having different sets of familiarity values, aged at different speeds around the actual aging speed, and periodically check which speed produces the best density of the graph and shift the actual aging speed in that direction.

### 4.3 Combining Metrics

Trust does not only come in different forms, but is also required to achieve different goals. Depending on the requirement, some forms of trust might be more suitable than others. Whereas social trust helps mostly to identify legitimate users, similarity trust is more useful to get valuable opinions. An adaption to the different needs can be done the following way. A trusted interaction partner is identified if

$$w_s \cdot t_s + w_e \cdot t_e \geq th_{lu} \quad (1)$$

with the weights  $w_s$  and  $w_e$  and the threshold  $th_{lu}$ . Since the amount of distributed social trust is higher than the environmental trust (see Section 5), a reasonable example is to have both weights of 1 and the threshold set to 0.1. A valuable opinion is identified by

$$(w_s \cdot t_s + w_e \cdot t_e) \cdot t_{ts} \geq th_{op} \quad (2)$$

with a taste similarity trust value  $t_{ts}$  in the range of  $[0, 1]$ . A reasonable value for the threshold  $th_{op}$  is 0.01. For an analysis of the amount of distributed trust and the implications of these suggested values see the next section.

## 5. EVALUATION

The algorithms proposed above are designed to work in a mobile setting, with a lot of interacting user, not all of them with good intentions. For this reason, it is important to know the complexity of establishing trust, understand how trust propagates in the graph and be aware of the security issues those algorithms imply. In order to analyze the algorithms, real world mobility traces where used, the MIT Reality traces [33] and the Huggle Infocom traces [34] to be more concrete. The traces consist of two totally different settings, a campus setting consisting of 96 nodes over a period of about 9 month with low contact density and a conference setting consisting of 41 nodes over a period of 3 days with high contact density.

### 5.1 Friendship Graph

*Complexity.*

Algorithm 1 consists of two parts which run each time a node comes into proximity. The parts consist of the construction of the friendship graph  $G_F$  and the calculation of the trust values  $t_s$ . The construction of  $G_F$  can be done by a modified BFS algorithm, thus has complexity  $O(b^d)$  with  $b$  being the branching factor and  $d$  being the depth of the resulting tree. The calculation of the trust values as stated in Algorithm 1 would take  $O(b^{2d-1})$  operations since for every trust value of a node in level  $d$ , all the trust values on level  $d - 1$  have to be summed up in the worst case. In order to optimize this, the trust values can be calculated on the fly when building  $G_F$ . This is shown in a more detailed version of the first loop of Algorithm 1 in Algorithm 3. By using this optimization the resulting algorithm has complexity  $O(b^d)$ . Additionally, the algorithm does not necessarily have to run after each time a new node was connected, especially if several nodes are in proximity at the same time. It is more efficient to exchange the friends list with several nodes and then run the algorithm just once.

As far as the size of the transmitted data is concerned it depends only on the amount of friends a node has, thus is in the order of  $O(b)$ .

*Trust Propagation.*

Algorithm 1 makes sure that each direct friend has a trust value of 1. Through the direct friends some of that trust propagates to their friends on the next level and so on. The amount of trust that propagates from one level to the next depends on the current level's total social trust  $T_d$  and the amount of nodes in the current level and the amount of edges going to the next level as described in Equation 3.

$$T_{d+1} = T_d \cdot \frac{|E_{d,d+1}|}{|L_d|^2} \quad (3)$$

---

**Algorithm 3** Friendship Graph

---

$n_i$ : A node (local node:  $n_0$ )  
 $e_{i,j}$ : Edge from  $n_i$  to  $n_j$   
 $ts_i$ : Social trust value of  $n_i$   
 $L_d$ : Set of nodes with distance  $d$  from  $n_0$  in  $G_F$   
 $c$ : Minimum degradation constant  
 $L_0 = n_0$   
 $size(L_0) = 1$   
 $ts_0 = 1$   
**for all**  $d \geq 0$  **do**  
  **for all**  $n_j$  in  $L_d$  **do**  
    **for all** unvisited  $e_{j,k}$  **do**  
      mark  $e_{j,k}$  as visited  
      **if**  $n_k$  not *visited*( $< d$ ) **then**  
        mark  $n_k$  as *visited*( $d$ )  
        add  $n_k$  to  $L_{d+1}$   
        increment  $size(L_d)$   
         $ts_k = ts_k + \frac{1}{\max(size(L_{d-1}), c^{d-1})} \cdot ts_j$   
      **end if**  
    **end for**  
  **end for**  
**end for**

---

trust type	MIT (96 nodes)			Haggle (41 nodes)		
	env.	soc. (sw/phone)	soc.+env. (sw/phone)	env.	soc.	soc.+env.
trust per node	1.94	7.98 / 4.75	9.91 / 6.68	1.59	7.48	9.07
nodes w/ trust $\geq 0.10$	5.9	15.4 / 9.1	21.0 / 14.7	2.9	14.4	17.9
nodes w/ trust $\geq 0.01$	22.6	38.4 / 16.9	53.4 / 35.7	23.8	29.7	36.1

**Table 1: Trust Value Statistics**

This results in

$$T_d = \prod_{i=1}^d \frac{|E_{i-1,i}|}{|L_{i-1}|^2} \quad (4)$$

as the total social trust for each level  $d$ . As a result, the overall propagated trust increases ( $T_d > T_{d-1}$ ) in case there are many edges between levels ( $|E_{d-1,d}| > |L_{d-1}|^2$ ). This is usually the case for the first few levels but changes rapidly as  $|L_{d-1}|$  increases since the average branching factor does not change over the levels. In order to assess the propagated trust, the algorithm has been tested on social graphs. Unfortunately both traces contain no information to base social trust on. For this reason, a synthetic small world graph is build and overlaid on the traces. The Watts and Strogatz model [35] is used to construct the graph. The  $n$  nodes of the network are arranged to a ring and friend ties are established with  $k = 4$  of their neighbors. Then, each link is rewired with a probability  $p = 0.25$  to a random node outside the  $k$  neighbors. Additionally, for the MIT set, a second friendship graph is constructed as a comparison. The links in the graph correspond to phone calls that took place. Although a phone call does not necessarily imply friendship, the resulting graph has nice social properties such as communities. The resulting trust propagation for the MIT Reality traces can be seen in Figure 1. The x-axis correspond to the hop distance in the friendship graph and the y-axis are the mean assigned trust values for nodes having that distance. The values are shown for both social graphs for two different degradation factors  $c$  of 1 and 2.

The values for the graph based on the phone records degrades slower over the hop distance. This is mainly because the average node degree in only 2.33, compared to the small world graph with average degree 4. The red dotted line marks the trust value of 0.1. We can see that most of the nodes achieving a trust value higher than 0.1 are under 3 hops away. Although it looks like in the phone graph the propagated trust is higher, this is actually the other way around. In the small world graph nearly double the trust propagates into a node on average. Also the number of nodes receiving a trust value over 0.01 and 0.1

is around double for the small world graph. The specific values<sup>2</sup> can be found in Table 1. Note that the values are very similar for the small world graph with 41 nodes overlaid on the Haggle traces. This suggests that the amount of propagated trust does not strongly depend on the network size but more so on the graph’s structure.

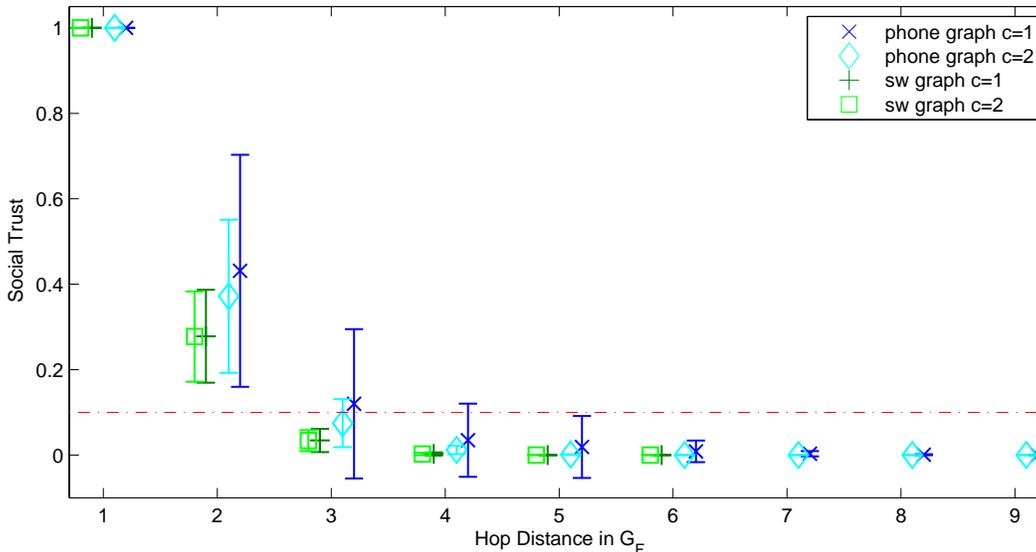


Figure 1: Mean Trust per Level in  $G_F$  of the MIT Data Set

*Security.*

Trust is a measure to establish security and for that reason, the establishment of trust has to be secure in the first place. Social trust is inferred from consciously established friend ties created by a user through a secure pairing process. It is up to the user to select and identify trustworthy friends. The resilience of this trust metric is thus up to the intelligence and reasonability of the user. Another way would be to compromise a device with malware. Malware is altogether an orthogonal problem which is beyond the scope of this paper.

Mistakes while estimating the trustworthiness of a friend or by installing malware are possible and do happen. For this reason the influence of a compromised node should be analyzed. On the one hand, a compromised node enjoys a certain trust from other nodes, and on the other, a compromised node can try to integrate sybil users into the friendship graph which then are trusted to some degree as well. From Equation 3 results that a node on a certain level of the friendship graph has only an influence on successive levels. Although the overall trust may increase from one level to the next, this is only possible if the branching factor increases in quadratic order with each level. This is only feasible for low levels in  $G_F$ , e.g. for direct friends, and can be minimized by limiting the maximum number of friends. This does not only make sense from a social point of view [36] but also from a computational perspective it helps to not blow the computation time out of proportion. Additionally, a more effective and also reasonable measure is to ignore all outgoing links of nodes that have a trust value below a certain threshold, as e.g. 0.01. This reduces the computational complexity by eliminating edges over which the trust propagation is negligible anyway.

**5.2 Environmental Examination**

*Complexity.*

For Algorithm 2 only one’s familiars and their familiars are relevant. The algorithm visits every familiar and their familiar and has thus a complexity of  $O(b^2)$ ,  $b$  being the branching factor or the average number of familiars for that matter. Although the complexity is much lower than for the social trust, assuming

<sup>2</sup>For these values  $c = 2$  is assumed.

$d > 2$ , it can still be computationally intensive if the number of familiars grows too much. In order to keep this aspect under control, an appropriate aging mechanism is necessary.

As far as data transfer is concerned, only the list of familiarity values has to be exchanged, thus the exchanged data is in the order of  $O(b)$

### *Trust Propagation.*

The total amount of trust that is distributed per node is at most 2. At least half of that belongs to direct familiars. Each familiar  $n_i$  receives  $\frac{f_{0,i}}{f_{s0}}$  amount of trust for itself and the same amount to distribute among its familiars. Although the amount of trust is limited, there is still more than 20 nodes that receive a trust value of at least 0.01 in both, the MIT as well as the Huggle data set. The average amount of nodes receiving a trust value or at least 0.10 is much smaller. The specific values can be found in Table 1 including for the case when social and environmental trust is combined.

In order to understand how Algorithm 2 distributes trust among the nodes in the local community, the trust values are mapped against the outcome of state of the art community detection algorithms. The Louvain algorithm [17] was used to find hierarchically organized communities by trying to optimize the modularity. The outcome can be seen in Figure 2(a). On the x-axis the corresponding hierarchical level of a node is shown. The y-axis shows the expected trust a node on that level. Nodes that are in the same community obviously have much higher trust values on average. The red dotted line marks the trust value of 0.1. We can see that nodes outside the community do not exceed this level of trust. The correlation improves if an aging mechanism is present. Although the dynamic aging<sup>3</sup> improves the distribution of the trust values, it can still be improved as shows the comparison to the static aging with good parameters<sup>4</sup> which is a matter of future work.

Although the correlation between the community membership and trust values are already visible, the environmental trust is not based on the local community with optimal modularity. For this reason, the simple algorithm [18] was used as well, for it is also based on the familiarity and similarity of the nodes. The results can be seen in Figure 2(b). The x-axis shows the classification of a node by the simple algorithm and the y-axis shows the expected trust the nodes receive. The simple algorithm was applied to the MIT traces using the optimal parameters the authors propose. Here the correlation of the trust values to a node's classification is even stronger but in both figures, 2(a) and 2(b), is seen how the distribution of the trust values among the local community reflects the structure of the network.

### *Security.*

The main goal of environmental trust is to increase the relevance<sup>5</sup> of a node and to make sure the node is not a fast switching identity. For this reason the process of assigning trust to nodes should be resilient to attacks. Algorithm 2 assesses trust of a node by the node's familiarity and by its familiarity to one's familiars. Both are properties that are not easily adulterated. In order to gain a lot of trust a node has to maximize its familiarity with the target nodes and minimize the familiarity other nodes might have with the target nodes. Since the only way to influence one's familiarity is by actually being present, an attacker either has to follow a target or to hide a node close to a place the target stays regularly. A strong antenna could additionally be used to increase the range of influence. In order to minimize other nodes familiarity, their beacons can be jammed. All these methods take a considerable effort and whether the influence on some nodes is worth the effort is questionable. Additionally, other sybil countermeasures [27, 28, 29] can be used in order to further increase the effort of an attack.

## 6. CONCLUSION

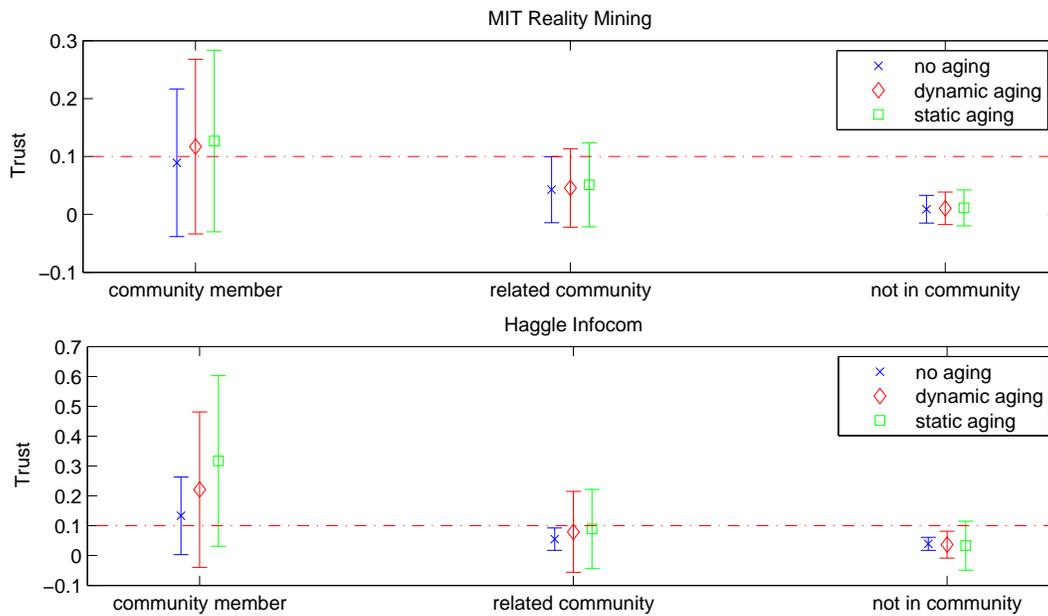
In the absence of a central authority (CA), the generation of many identities by a single user is possible. While a true one-to-one, once-in-a-lifetime binding between entity and identity is already challenging with a central authority, it becomes even more so in a fully distributed and more specifically opportunistic environment. Sybil attacks and the ease of new identity generation raises the need of solid trust metrics.

---

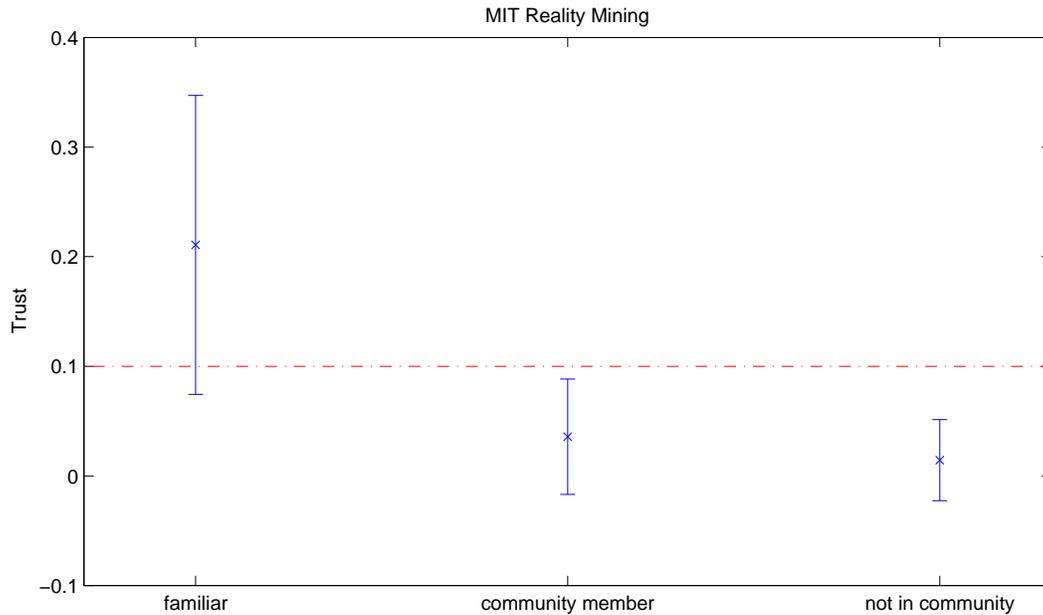
<sup>3</sup>Dynamic aging is a prototype aging mechanism based on optimizing density of the network (see Section 4).

<sup>4</sup>The parameters chosen were an aging speed of  $1 \frac{s}{h}$  for the MIT Reality traces and  $100 \frac{s}{h}$  for the Huggle Infocom traces.

<sup>5</sup>The frequent encounter makes the node an important interaction partner, e.g. because unfinished transmissions can be finished in the near future.



(a) Louvain Hierarchy



(b) Simple Classification

**Figure 2: Distribution of Trust Values**

For this reason we make use of several forms of trust, namely *social*, *environmental* and *similarity trust*, to satisfy the different requirements an application based on an opportunistic network might have. Trust is mainly assessed for the part of the network which is relevant to a node, which makes this approach scalable. Of course, not all information about every encountered user can be saved and a true one-to-one binding of entity and identity cannot be guaranteed. This does not diminish the validity of this approach, since most of the application do not care about the entity behind the identity but only the trust relationship between identities.

## 7. REFERENCES

- [1] "The aka aki network." <http://www.aka-aki.com/>.

- [2] G. Karlsson, V. Lenders, and M. May, "Delay-tolerant broadcasting," *IEEE Transactions on Broadcasting*, vol. 53, pp. 369–381, March 2007.
- [3] S. Gaonkar, J. Li, R. R. Choudhury, L. Cox, and A. Schmidt, "Micro-blog: sharing and querying content through mobile phones and social participation," in *Proceeding of the 6th international conference on Mobile systems, applications, and services (MobiSys)*, (New York, NY, USA), pp. 174–186, ACM, 2008.
- [4] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Computing Surveys*, vol. 35, no. 2, pp. 114–131, 2003.
- [5] J. Sabater and C. Sierra, "Social ReGreT, a reputation model based on social relations," *SIGecom Exch.*, vol. 3, no. 1, pp. 44–56, 2002.
- [6] S. Capkun, J.-P. Hubaux, and L. Buttyan, "Mobility helps peer-to-peer security," *IEEE Transactions on Mobile Computing*, vol. 5, pp. 43–51, January 2006.
- [7] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, no. 2, pp. 167–256, 2003.
- [8] J. Liu and V. Issarny, "Enhanced reputation mechanism for mobile ad hoc networks," in *Proceedings of the 2nd International Conference on Trust Management (iTrust)* (C. D. Jensen, S. Poslad, and T. Dimitrakos, eds.), vol. 2995 of *Lecture Notes in Computer Science*, pp. 48–62, Springer, 2004.
- [9] K. Walsh and E. G. Sirer, "Experience with an object reputation system for peer-to-peer filesharing," in *Proceedings of the 3rd conference on 3rd Symposium on Networked Systems Design & Implementation (NSDI)*, (Berkeley, CA, USA), Networked System Design and Implementation (NSDI), USENIX Association, May 2006.
- [10] R. Housley, W. Ford, W. Polk, and D. Solo, "Internet x.509 public key infrastructure certificate and crl profile," 1999.
- [11] Y.-H. Lin, A. Studer, H.-C. Hsiao, J. M. McCune, K.-H. Wang, M. Krohn, P.-L. Lin, A. Perrig, H.-M. Sun, and B.-Y. Yang, "SPATE: Small-group PKI-less Authenticated Trust Establishment," in *Proceedings of the 7th international conference on Mobile systems, applications, and services (MobiSys)*, (New York, NY, USA), pp. 1–14, ACM, 2009.
- [12] S. Capkun, L. Buttyan, and J.-P. Hubaux, "Self-organized public-key management for mobile ad hoc network," *IEEE Transactions on Mobile Computing*, vol. 2, pp. 52–64, January-March 2003.
- [13] P. R. Zimmermann, *The Official PGP User's Guide*. MIT press, 1995.
- [14] D. Liben-Nowell and J. Kleinberg, "The link-prediction problem for social networks," *Journal of the American Society for Information Science and Technology*, vol. 58, no. 7, pp. 1019–1031, 2007.
- [15] E. M. Daly and M. Haahr, "Social network analysis for routing in disconnected delay-tolerant manets," in *Proceedings of the 8th ACM international symposium on Mobile ad hoc networking and computing (MobiHoc)*, (New York, NY, USA), pp. 32–40, ACM Press, 2007.
- [16] M. E. J. Newman, "Fast algorithm for detecting community structure in networks," *Physical Review E*, vol. 69, p. 066133, 2004.
- [17] V. D. Blondel, J.-L. Guillaume, R. Lambiotte, and E. Lefebvre, "Fast unfolding of communities in large networks," *Journal of Statistical Mechanics: Theory and Experiment*, vol. 2008, no. 10, p. P10008 (12pp), 2008.
- [18] P. Hui, E. Yoneki, S. Y. Chan, and J. Crowcroft, "Distributed community detection in delay tolerant networks," in *Proceedings of 2nd ACM/IEEE international workshop on Mobility in the evolving internet architecture (MobiArch)*, pp. 1–8, ACM, 2007.
- [19] P. Hui, J. Crowcroft, and E. Yoneki, "Bubble rap: Social-based forwarding in delay tolerant networks," in *Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc)*, pp. 241–250, ACM, 2008.
- [20] T. Hossmann, T. Spyropoulos, and F. Legendre, "Know thy neighbor: Towards optimal mapping of contacts to social graphs for dtn routing," in *IEEE Infocom*, 2010.
- [21] S. Chan, P. Hui, and K. Xu, "Community detection of time-varying mobile social networks," in *Complex Sciences*, Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer, February 2009.
- [22] S. Buchegger and J.-Y. Le Boudec, "A Robust Reputation System for Peer-to-Peer and Mobile Ad-hoc Networks," in *Proceedings of the Second Workshop on the Economics of Peer-to-Peer Systems (P2PEcon)*, 2004.
- [23] D. Quercia, S. Hailes, and L. Capra, "B-trust: Bayesian trust framework for pervasive computing," in *Proceedings of the 4th IEEE International Conference on Trust Management (iTrust)*, pp. 298–312, Springer Verlag, 2006.
- [24] D. Quercia, S. Hailes, and L. Capra, "Lightweight distributed trust propagation," in *Proceedings of the 7th IEEE International Conference on Data Mining (ICDM)*, (Washington, DC, USA), pp. 282–291, IEEE Computer Society, 2007.
- [25] D. Quercia, S. Hailes, and L. Capra, "MobiRate: Making Mobile Raters Stick to their Word," in *Proceedings of the 10th international conference on Ubiquitous computing (UbiComp)*, (New York, NY, USA), pp. 212–221, ACM, 2008.
- [26] J. R. Douceur, "The sybil attack," in *Revised Papers from the First International Workshop on Peer-to-Peer Systems (IPTPS)*, vol. 2429 of *Lecture Notes in Computer Science*, (London, UK),

- pp. 251–260, Springer-Verlag, March 2002.
- [27] C. Piro, C. Shields, and B. N. Levine, “Detecting the sybil attack in mobile ad hoc networks,” in *Securecomm and Workshops*, pp. 1–11, September 2006.
  - [28] H. Yu, M. Kaminsky, P. B. Gibbons, and A. D. Flaxman, “Sybilguard: Defending against sybil attacks via social networks,” *IEEE/ACM Transactions on Networking*, vol. 16, pp. 576–589, June 2008.
  - [29] A. Tangpong, G. Kesidis, H. yuan Hsu, and A. Hurson, “Robust Sybil Detection for MANETs,” in *Proceedings of 18th International Conference on Computer Communications and Networks (ICCCN)*, pp. 1–6, August 2009.
  - [30] D. Gambetta, *Can We Trust Trust?* Basil Blackwell, 1988.
  - [31] G. Swamynathan, C. Wilson, B. Boe, K. Almeroth, and B. Y. Zhao, “Do social networks improve e-commerce?: A study on social marketplaces,” in *Proceedings of the first workshop on Online social networks (WOSN)*, (New York, NY, USA), pp. 1–6, ACM, 2008.
  - [32] A. Clauset, “Finding local community structure in networks,” *Physical Review E*, vol. 72, no. 2, p. 026132, 2005.
  - [33] N. Eagle and A. (Sandy) Pentland, “Reality mining: sensing complex social systems,” *Personal and Ubiquitous Computing (PUC)*, vol. 10, no. 4, pp. 255–268, 2006.
  - [34] A. Chaintreau, P. Hui, C. Diot, R. Gass, and J. Scott, “Impact of human mobility on opportunistic forwarding algorithms,” *IEEE Transactions on Mobile Computing*, vol. 6, no. 6, pp. 606–620, 2007. Fellow-Crowcroft, Jon.
  - [35] D. J. Watts, *Small Worlds : The Dynamics of Networks between Order and Randomness*. Princeton University Press, 2003.
  - [36] R. Dunbar, “Coevolution of neocortex size, group size and language in humans,” *Behavioral and Brain Sciences*, vol. 16, no. 4, pp. 681–735, 1993.