Trust Is in the Eye of the Beholder

Dimitri DeFigueiredo* Earl Barr† S. Felix Wu†

*Adobe Systems Inc.
defigueiredo@ucdavis.edu

†University of California, Davis
{etbarr,sfwu}@ucdavis.edu

Abstract

We carefully investigate humanity's intuitive understanding of trust and extract from it fundamental properties that succinctly synthesize how trust works. From this detailed characterization we propose a formal, complete and intuitive definition of trust.

Using our new definition, we prove simple possibility and impossibility theorems that dispel common misconceptions, expose unexplored areas in the design of reputation systems and shed new light on the shortcomings of previous impossibility results.

Keywords: trust, trust transitivity, reputation systems, non-exploitability, personalized reputation function.

1 Introduction

Many internet applications use trust to help users in their online interactions with others. The Feedback Forum, eBay's rating system, is perhaps the best known example. Trust, in different flavors, is particularly useful to distributed online applications such as peer-to-peer or social networks [10]. Both online and off-line, knowing how much to trust someone helps us know what to do in our interactions with them. To better understand the concept of trust online, we begin exploring mundane off-line real-world trust, the concept we seek to mimic.

Trust systems use reputation functions to quantify trust. Consensus-based reputation functions have intrinsic limitations, as we show in Theorem 3.1. In particular, they are exploitable — they can be manipulated by untrusted parties. We introduce a new class of reputation functions, *personalized* reputations functions that overcome these limitations. Personalized reputation functions allow a person to control how much she trusts another party, independent of the opinion of others, thus allowing her to make her own, individual, assessment of each party's trustworthiness. Consensus-based reputation functions require both Alice and Bob to agree on how trustworthy Charlie is. Personalized reputation functions do not.

To date, most trust systems use reputation functions that arbitrarily map trust to numbers. We quantify trust in terms of utility. This insight links the trust transitivity problem, *i.e.* how much should you trust the friend of a friend, to the problem of making interpersonal comparisons of utility.

People confuse trust and trustworthiness: they tend to think how much they trust a person is an objective measure of that person's trustworthiness, that everyone would agree with their trust assessment if only they "knew what I know." Trust is, in fact, independent of trustworthiness. Alice may trust Bob even though he is not trustworthy; Alice may distrust Charlie even though he *is* trustworthy. This confusion underlies the prevalence of trust systems built on consensus-based reputation functions. Below, we draw a bright line between trust and trustworthiness, setting the stage for trust systems built on personalized reputation functions.

We make the following contributions:

1. We introduce and formalize personalized reputation functions;

2. We provide a definition of Trust that exposes the link between trust transitivity and interpersonal comparisons of utility; and

3. We elucidate the difference between trust and trustworthiness.

This paper is structured as follows: In Section 2, we formalize a notion of trust that is domain-specific and captures the difference between trust and trustworthiness. Our formalism also shows the link between the transitivity of trust and interpersonal comparisons of utility. In Section 3, we provide possibility and impossibility results. We show that all consensus-based reputation systems are exploitable by untrustworthy parties, but also that there are non-exploitable personalized reputation systems, such as TrustDavis which we have previously proposed [4]. The latter result dispels fallacies from previous impossibility theorems [3]. Section 4 discusses the implications of these results in relation to previous work. Section 5 concludes.

2 Formalizing Trust

Trust involves two roles — a *trusting* and a *trusted* role. Distinct parties usually perform these roles.

Generally, trust is domain-specific: we may trust a person in one domain, but not in another. For example, I trust my father to take care of my finances, but I do not trust him to be on time for dinner. We tend to aggregate disparate domains such as "being on time" or "handling money" into more general ones. In fact, I could say: "I trust my father," independent of domain. In these aggregate domains, we may not be able to decide whether we trust our friend more than our sister! These abstract domains may be partially ordered. However, if given a specific domain, we can usually provide a more specific assessment of trustworthiness. In what follows, we assume specific domains that are totally ordered.

Definition 2.1 (trust values)**.** Trust values are real numbers.

This definition implies that trust is quantifiable on a single scale and that, from a single agent's point of view, there is a total ordering when comparing how trustworthy different agents are. If Alice tells us that (for the same domain):

- Alice trusts Bob more than Charlie;
- Alice trusts Charlie more than Derek; and,
- Alice trusts Derek more than Bob.

We believe she would be willing to change her mind once we point out the inconsistency.

We do not make further restrictions such as assuming that trust is a binary variable with two possible values $(-1, 1)$ or that it is in the interval $[0, 1]$, except that we do require that all agents agree that higher values are better (see the definition of trust threshold below).

We would like to point out a useful interpretation for trust values. One can think of trust values as being specified in dollars. Positive values answer the question: How much money are you willing to risk on this person? The higher the value, the more trusted the person is. This leads to a similar interpretation for negative trusts values. If v_1 trusts v_2 a negative amount $-x$, then x specifies how much money v_1 would have to risk gaining (not losing) to be worth the risk of depending on the behavior of v_2. This interpretation motivates the definition of trust we provide later in this section.

Definition 2.2 (reputation graph)**.** A *reputation graph* is an annotated directed Graph $G = (V, E)$, where each vertex is an agent and each directed edge in E has an associated trust value. The reputation graph does not need to be complete.

We interpret this reputation graph G as follows. There is a directed edge labeled x from vertex v_1 to vertex v_2, if v_1 trusts v_2 the specified amount x. If there is no edge between two vertices, then the amount of trust between them is unspecified.

We could build such a graph by asking each agent about all others. For example, we could ask v_1: How much are you willing to bet that v_2 is a good cook? And then add an edge to the graph, starting at v_1 and ending at v_2, labeled by the corresponding dollar amount. If v_1 does not know how much he trusts v_2,

we would not add an edge. In light of this interpretation, we consider all of a vertex' outgoing edges to be data local to and under the control of that vertex.

The next definition accomodates the transitivity of trust. Specifically, the degree to which people use second-hand opinions when considering a domain. For example, although I trust Bob to be honest, he is naive and therefore I do not trust his judgement about the honesty of others.

Definition 2.3 (world)**.** A *World* is a sequence of reputation graphs
$W = \{G_k = (V, E_k), k \geq 0\}$, where all reputation graphs use the same set of vertices. We call the first reputation graph G_0 the *direct experience graph*. The following graphs are called the *1-indirect graph*, the *2-indirect graph*, and so on.

Let us use the domain of "being a good cook" as an example. When building the direct experience graph one should only use information obtained from actually eating the food cooked by another agent. No information received indirectly from other parties should be used. There should only be an edge going from v_1 to v_2 if v_1 actually tasted food prepared by v_2 and has an opinion on it. Similarly, the 1-indirect graph should answer the question: Is this person a good food critic? In other words, Does v_1 trust v_2's palate to evaluate someone else's cooking? The sequence of graphs represents increasing levels of indirection for the same initial domain.

A world has a single set of vertices, but otherwise the graphs G_k can change arbitrarily for different k. Each graph has a different set of edges E_k, each labeled with different trust values. The definition of a world mirrors the multiple domains of trust and that there may not be correlations between these domains. Our results hold regardless of whether or not there exist correlations between edges in different graphs: the definition takes into account settings where some agents trust their cooks as good food critics and other agents do not.

Definition 2.4 (reputation function)**.** Given a world, a *reputation function* f assigns a trust value to each ordered pair of vertices.

Definition 2.5 (trust graph)**.** A reputation function outputs a *complete,* directed *trust graph*.

The reputation function tells us how trustworthy v_1 thinks v_2 is, for every pair (v_1, v_2) in the world. Because two parties may not have interacted in the past, we let a reputation graph be incomplete. However, because an agent does not know whom it will meet next and may need to suddenly form an opinion, we require the trust graph to be complete. Trust graphs and reputation graphs represent different concepts. The reputation graphs represent the reputation information available. Given that information, the trust graph expresses how trustworthy each party is, from different points-of-view. We will consider the trust graph in detail later. Given the world W, we use $f[W](v_i, v_j)$ to denote the trust value f assigns to the edge (v_i, v_j) in the trust graph. When W is clear from context, we simply use $f(v_i, v_j)$.

We require that all the reputation information available to a reputation function be expressed in the edges and edge labels of the different graphs that constitute a world. No other information should be used. This can be made precise by the following isomorphism requirement:

Definition 2.6 (normalization)**.** Let us denote by πW the world obtained by permuting the vertices of W with permutation $\pi : V \to V$. A *normalized reputation function* f is one where $f[W](v_i, v_j) = f[\pi W](\pi(v_i), \pi(v_j))$ for any permutation of the vertices π and for any world W.

The above definition simply states that if we permute the input to the reputation function we should obtain the same permutation of the output. This requirement forces reputation functions not to obtain extra information from their input in the form of vertex labels. For example, assume that v_2 trusts v_1 because v_1 represents the *New York Times.* In other words, v_1 has the vertex label *"New York Times"*. Why does v_2 trust the *New York Times*? Is it not because of the New York Time's *reputation*? If so, this reputation information should be added to the world as edges (v_2, v_1) with the corresponding edge labels. Note that it is easy to transform vertex labels into edges that convey the same information.

Normalization forces any knowledge about the world to be made explicit in the edges and edge labels of the world given as input to the reputation function. This requirement also prevents reputation functions that perform better only when there are distinguished nodes from being mistaken for functions that work well in more general settings. It does not prevent such functions from obtaining information, it just makes the information used explicit. The normalization helps prevent the designer from comparing apples to oranges when comparing two reputation functions. For the remainder of this paper, we put all reputation information on edge labels and will restrict our attention to normalized reputation functions.

Definition 2.7 (consensus-based vs. personalized)**.** A *consensus-based reputation function* is one where, for all vertices v_j in the world graph, the trust values x_{ij} assigned to ordered pairs of vertices (v_i, v_j) are the same for all pairs ending in the same vertex v_j. Therefore, the trust value $x_{1j} = x_{2j} = ... = x_{nj} = y_j$ assigned by the reputation function is completely determined by v_j and denotes v_j's trustworthiness to all other agents. A reputation function that is not consensus-based is *personalized*.

In effect, a consensus-based reputation function assigns a trust value to each vertex in the trust graph it outputs, instead of assigning a trust value to each edge in that same graph.

Definition 2.8 (trust threshold)**.** The *trust threshold* for agent v_i is a trust value h_i established by that agent. All agents v_j whose trust values for (v_i, v_j) assigned by the reputation function are above the threshold h_i are **trusted** by v_i; otherwise they are **untrusted**. In other words, if $f(v_i,v_j) \geq h_i$ then v_i trusts v_j; otherwise v_i does *not* trust v_j.

Note that this classification of agents as trusted and untrusted is very flexible. It allows agents with negative trust values to be trusted or agents with positive trust values to be untrusted depending on the trust threshold each trusting party chooses.

We can now also define what we mean by distrust. In everyday life, we can usually classify others as trusted, neutral or distrusted. This leads to a simple definition of distrust: all agents strictly below the trust threshold are distrusted; *i.e.* , if $f(v_i, v_j) < h_i$ then v_i distrusts v_j. This is a very simple notion, but we have already captured it with our definition of an untrusted agent above.

We believe there are two alternative, equally intuitive, possible definitions of this concept. One is that an agent v_j is distrusted by the trusting party v_i if $f(v_i, v_j) < 0$. According to our previously suggested interpretation of trust values, the trusting party needs to be tempted by the possibility of gain in order for him to interact with a distrusted party. Another definition is that *a node is distrusted whenever it is trusted less than a complete stranger*[1]. This definition implicitly assumes that all complete strangers are trusted to the same extent by the trusting party v_i. We call this the *stranger threshold* s_i.

Most of time the stranger threshold is zero and the definitions are equivalent and equally appealing. However, it is possible for one to be trusting of strangers and thus willing to take risks that depend on their behavior. This would be analogous to setting a high stranger threshold, *i.e.* $s_i > 0$. This does not necessarily mean that strangers are trusted (as that requires that $s_i \geq h_i$ and it may be that $0 < s_i < h_i$), but only that the trusting party is willing to take some risks based on their behavior.

We adopt the latter definition for distrust. We do so because it brings out the observation that distrust is useful, mostly when parties cannot easily get new identities. Otherwise, it is easy for malicious players to issue themselves a new identity everytime they want to fool somebody. The "cheap pseudonym problem" [5] rears its head again.

We are now ready to provide a definition of trust.

Definition 2.9 (Trust)**.** *Trust* is the personal threshold determined by the trusting party that describes the maximum utility the trusting party is willing to risk when dealing with the trusted party.

The definition quantifies trust in terms of utility and is a significant contribution of this paper. Expressing trust in terms of utility makes the link between

[1]For brevity we omit the formal definition of complete strangers.

the trust transitivity problem and interpersonal comparisons of utility very clear. The trust transitivity problem is a consequence of and reduces to the problem of making interpersonal comparisons of utility. The last two definitions also make precise colloquial questions like: "How much do you trust him?" and "Is he trustworthy?"

Definition 2.10 (collusion)**.** A *collusion* is a subset of the vertices of a World.

Definition 2.11 (untrusted collusion)**.** An *untrusted collusion*, from agent v's perspective, is a collusion whose members are *all* untrusted agents.

Definition 2.12 (manipulated world)**.** Given a world $W = \{(V, E_k)\}$ and a collusion $C \subseteq V$, a *world manipulated by that collusion* $W'_C = \{(V', E'_k)\}$ is a modified world in which the collusion can change the reputation graphs in two ways:

- The collusion can arbitrarily add or remove edges starting from any member of the collusion; *i.e.,* for any $v_i \in C$ and $v_j \in V$, we can add or remove any edge of the form (v_i, v_j) in any reputation graph.
- The collusion can arbitrarily change the trust values on any edge starting from any member of the collusion (including any edges they have added).

In all other respects W'_C is identical to W. In particular, $V' = V$.

A non-exploitable reputation function is one for which each trusting party can determine autonomously and arbitrarily how much she trusts others. This implies that no untrusted collusion can increase the trust value of any agent. This enables the trusting party to control how much trust she places on others and to set an absolute bound on how much damage any collusion of malicious agents can do. The trusting party can still take risks if she so desires.

Definition 2.13 (non-exploitability)**.** A *non-exploitable reputation function* is a reputation function where: For any given world W, *for any vertex* v_i and any trust threshold h chosen by that vertex, no **untrusted** collusion in W can change the trust value of any agent v_j, *i.e.* $f(v_i, v_j)$. This is done by comparing trust values in two worlds: the standard honest world W and *any* world manipulated by an untrusted collusion.

Notice that this definition has six quantifications. We quantify over all worlds, for each world over all agents (trusting party role) and any trust threshold that agent may choose. Then, for each agent, for all possible collusions. For each such collusion, we consider all possible manipulations due to that collusion. Finally, for each possible manipulation we look at all agents (trusted party role) and ask whether that manipulation was able to change that agent's trust value. The quantifications are:

- over all worlds
- over all trusting parties and their thresholds
- over all collusions
- over all manipulations that collusion can perform
- over all agents whose values may be affected by these manipulations.

This definition encompasses *whitewashing* attacks [6], where an agent sheds a bad reputation and reappears as a newcomer, by its first quantification over all worlds. This quantification includes a world where both the new and old identities exist and collude with each other. Similarly, *sybil* attacks [3], where a malicious agent obtains multiple identities, are modelled as collusions.

A trivial reputation function is one that completely ignores the transitivity in trust and only takes into account local data.

Definition 2.14 (triviality)**.** A *trivial reputation function* f is a reputation function such that for any given world and for any pair (v_i, v_j) the value of $f(v_i, v_j)$ depends only on edges starting at v_i and is independent of all other edges.

This definition is actually slightly more general than required by our proofs. We are only concerned about

ruling out a reputation function that is fixed. We could have opted for a more restrictive definition as in [3] or [2]. However, we believe that triviality also applies to reputation functions that use only local, but not necessarily fixed, information. That being so, triviality is better defined as above.

3 The Limitations of Trust

Our impossibility result follows clearly from the definitions above.

Theorem 3.1. *All non-exploitable consensus-based reputation functions are trivial.*

Proof. We assume that a consensus-based reputation function f is non-trivial and show that it is exploitable.

If there is only one node, all information is local and all reputation functions satisfy the condition for triviality, therefore we will assume that there exist at least two nodes. Assuming we have a reputation function that is consensus-based, any trusted party has a single universal trust value. Consider any two distinct nodes v_1 and v_2. We define two untrusted collusions — U_1 (untrusted by v_1), and U_2 (untrusted by v_2). The set of all nodes is V and $\overline{U}_1 = V - U_1$ is the set of all nodes not in U_1. In the trust graph output by the consensus-based reputation function f using the honest world W as input, v_1 has a trust value of t_1 and v_2 has a trust value of t_2.

Assume that U_1 now wants to trick the trusting party v_1 and change t_2. There are only two possibilities: either U_1 is able to change t_2 or not. If U_1 can change t_2, then f is exploitable, by definition. We therefore continue with the assumption that U_1 cannot change t_2, and consider the situation from v_2's perspective and ask "Can $\overline{U}_1$ change t_2?"

As with U_1, $\overline{U}_1$ can either change t_2 or not. If $\overline{U}_1$ cannot, f is trivial, since, if neither U_1 nor $\overline{U}_1$ can change t_2, t_2 is fixed (whomever v_2 might be) and we have contradicted our assumption that f is nontrivial. We therefore continue with the assumption that $\overline{U}_1$ can change t_2, and show that there exists a world in which $\overline{U}_1 \subseteq U_2$.

If $v_2 \in U_1$, it cannot change t_2, since we have assumed U_1 cannot change t_2. If $v_2 \notin U_1$ and it can change t_2, this fact does not make v_1 exploitable since v_1 trusts v_2: the question is whether v_2 can be exploited. Regardless of whether $v_2 \in U_1$ or not, v_2 can change h_2, its trust threshold. In particular, v_2 can set h_2 to be greater than the greatest trust value of any node in V. In so doing, v_2 sets $U_2 = V$ and no node is trusted. Thus, $\overline{U}_1$ is an untrusted collusion in v_2's eyes and f is exploitable against v_2.

So we have shown f is exploitable against v_1, or, if it is not, f either is trivial, in contradiction of our assumption that f is nontrivial, or f is exploitable against v_2. Therefore, f is exploitable. □

This theorem does not limit all reputation functions, only consensus-based ones. However, it does show that all non-trivial consensus-based reputation functions are exploitable. In other words, consensus-based reputation systems are intrinsically insecure. There is always a way to manipulate such systems.

Theorem 3.2. *There are non-trivial non-exploitable personalized reputation functions.*

The proof that follows is constructive. We build a binary (trusted/untrusted) reputation function that expresses a simple transitivity notion: "I trust you, if I trust somebody who trusts you". Without loss of generality, we set a global "reputation threshold" λ, but this is distinct from and does not preclude agents from picking their own trust thresholds. However, the binary nature of this toy function f_λ means that trust thresholds outside the interval $(0, 1)$ lead to either everyone being trusted or untrusted, whereas all threshold settings within the interval are mutually indistinguishable. The function f_λ also assumes that all reputation graphs are identical and disregards all but the direct experience graph. This is not as unrealistic an assumption as it may seem at first: if you are a good cook you can probably identify other good cooks. We proposed a reputation system based on a non-trivial, non-exploitable reputation function in previous work [4].

Proof. From lemmas 1 and 2 below we obtain that the function f_λ is non-trivial and non-exploitable. □

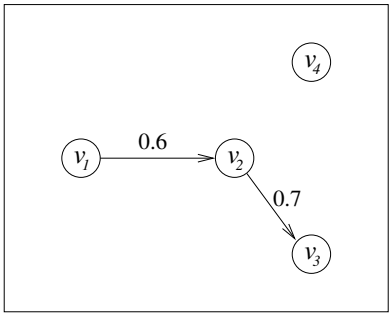

Figure 1: An example direct experience graph.

Definition 3.1 (trusted path)**.** Given a world W, a *trusted path* from vertex v_i to vertex v_j exists if there is a directed path in the direct experience graph from v_i to v_j along which all edges are labeled by at least λ.

Define f_λ as follows:

$$f_\lambda(v_i, v_j) = \begin{cases} 1, \text{ if a trusted path from } v_i \text{ to } v_j \text{ exists} \\ 0, \text{ otherwise} \end{cases}$$

Lemma 3.3. *The function f_λ is non-trivial.*

Proof. Let us set *w.l.o.g.* the reputation threshold $\lambda = 0.5$. Consider the direct experience graph with four nodes v_1,v_2, v_3 and v_4 depicted in Figure 1. The node v_4 has no incoming or outgoing edges; whereas v_3 can be reached from v_1 through the trusted path $(v_1, v_2), (v_2, v_3)$. The only asymmetry between v_3 and v_4 in the graph is that there is no edge (v_2, v_4) above the threshold λ while the edge (v_2, v_3) exists and is above the threshold. This data is not local to v_1 (it is local to v_2), but, because of the data at v_2, f_λ outputs different trust values making $f(v_1, v_3) = 1$ whereas $f(v_1, v_4) = 0$. Therefore f_λ depends on non-local data and is non-trivial. □

Lemma 3.4. *The function f_λ is non-exploitable.*

Proof. Consider a node v_i. If v_i's trust threshold is outside the interval $(0, 1)$ then either all nodes are trusted or untrusted. If all nodes are trusted, there can be no untrusted collusion and f_λ is (vacuously) non-exploitable. Similarly, if v_i's trust threshold is larger than one, then all nodes are untrusted and no collusion can change that outcome. If either holds for any node v_i, f_λ is non-exploitable. On the other hand, if v_i's trust threshold is within $(0, 1)$ then v_i trusts a node v_j if there is a trusted path from v_i to v_j. Because no trusted path to an untrusted node exists, untrusted nodes cannot add nodes to a trusted path. Similarly, because untrusted nodes are never in a trusted path to any trusted node (otherwise they would themselves be trusted), they cannot make a node unreachable either. From v_i's perspective, no untrusted collusion can change trust values and this is true for any node v_i. Therefore, f_λ is non-exploitable. □

The binary nature of the toy reputation function f_λ implies that trust from different paths along a reputation graph cannot be merged. If many vertices have trivial interactions with v_i, this function cannot combine the many small transactions into a single rating. It also does not capture the observation that trust grows with time.

Trust builds up over time with repeated successful interactions. Every successful interaction increases the reputation information available and those small increments slowly increase the trust ratings. Each vertex has a different adversity to risk and will increase the reputation of others differently after every interaction. However, it is possible to prudently increase those ratings while simultaneously decreasing one's risk exposure by capturing part of the trust flow along different paths.

Take eBay's reputation system as an example. It uses a single global trust value for each party and is therefore consensus-based. Our impossibility theorem shows that, because of this limitation, eBay's system can be manipulated by untrustworthy parties. At the same time, our possibility theorem shows that, if the ratings were displayed only after a user logged in, the system could be personalized for that user and overcome this limitation. For a non-trivial system, such as eBay's, the *only* way to prevent manipulation is to personalize the system for each user.

We proposed just such a reputation system, called TrustDavis, for online markets like eBay's [4]. TrustDavis realizes a non-trivial, non-exploitable reputation function. We refer the reader to that text for a complete description of a non-binary reputation function that monetizes trust values.

4 Discussion and Related Work

Reputation systems [14, 13] and the related topic of "trust management" have received a lot of attention from the computer science community. A good survey on trust from this point of view can be found at [10].

There is a close relationship between reputation functions and reputation systems [14]. To be of practical use, reputation systems must aggregate information about each agent's past history into an easy-to-use format. It is usually easier to perform a *single* aggregation for all users. However, this is not a requirement. The view that producing a single set of aggregation results for all users is the only way to aggregate the past history is precisely the *single worldview fallacy*. We do not imply that a single global worldview is not useful; to the contrary, it is very useful. However, it has limitations that can be overcome if one personalizes the results obtained (See Definition 2.7). Furthermore, personalizing the results obtained does not mean discarding information. Ranking systems such as the ones considered in [2] can be changed to provide personalized results for many e-commerce applications and multi-agent systems. There need not be a single ranking of alternatives, each party should be allowed to have their own preferences.

Non-exploitability just means that the trusting party is able to arbitrarily control how much trust she places on others, no matter what world setting she faces. This is an important characteristic of real-world trust that is captured by non-exploitable reputation functions. Non-exploitable reputation functions and systems output personalized worldviews, one view for each agent. Non-exploitability is what makes them more useful and easier to understand. How would you use trust if you could not control how much trust you could place in others? This important property has been overlooked by the literature.

Our definition of normalization (Definition 2.6) does not limit our characterization of trust. Cheng and Friedman partition reputation functions into symmetric, those whose output is solely dependent on edges, or actual interactions, and asymmetric, those whose output is computed with respect to a distinguished node [3]. The isomorphism property we use to define normalization is analogous to the isomorphism property presented in [2] and the definition of a symmetric reputation function. Symmetric functions are subject to Cheng and Friedman's impossibility result. Asymmetric functions rely on nodes labels that encode extrinsic information. In other words, asymmetric functions correctly encode trust only if they were correctly constructed from previously known trust information. This reputation-bootstrapping [11] is a form of the chicken and egg problem. Normalization avoids precisely this problem. The importance of normalization should not be played down because it seems too restrictive [3]: it is an essential requirement.

Reputation functions that are not normalized can be easily mapped to normalized functions. Our results hold given the appropriate mapping of the different domains. Furthermore, our possibility theorem contradicts previous results. Specifically, Cheng and Friedman show that "There is no symmetric sybil-proof nontrivial reputation function" because they did not consider personalized reputation functions [3]. Here, we show that there do exist normalized non-exploitable nontrivial *personalized* reputation functions. Elsewhere, we have shown that it is possible to build a practical trust system from such reputation functions [4]. Cheng and Friedman's misconception would be hard to dispel without the intuitive understanding of trust developed in this paper.

There is a large body of literature on trust from a variety of disciplines. Alternative definitions in a computational setting have been proposed for both trust [12] and distrust [8], and an interesting attempt to classify all the different points of view can be found in [1]. Our definition is closely related to the one found in the encompassing work of Gambetta [7]. However, our definition not only observes the link between trust and risks that depends on the behavior of others, but also makes clear the fundamental connection between trust and utility. This link casts light in the trust transitivity problem and shows how trust can be monetized.

Further testament to the usefulness of the formal definitions proposed comes from their breadth. Our definition of a world provides a very flexible trust transitivity framework and encompasses many of the settings found in the trust propagation (or transitivity)

literature [4, 9, 8, 15]. Despite the opposing results, we feel that the aims of our work most closely resemble those of [2] and [3].

5 Conclusion

Humans have an intuitive understanding of trust and are very proficient at using it as a tool to help them in their daily interactions with others. To make online interactions easier for users, a number of internet websites and peer-to-peer networks provide systems that explicitly attempt to capture the concept of trust. However, many of the systems in use today assign a single universal trust rating to each participating party. This implies that these systems are inherently vulnerable to manipulation by malicious users and are, therefore, not as useful to the end users as they could be.

Non-exploitable systems that do not provide a single universal trust rating, but that can change the trust ratings assigned to an individual, depending on domain and on who is asking the question, resist malicious manipulation. These systems are inherently more intuitive than the systems currently in use, and will become more useful tools to end users who seek help in dealing with the online world.

References

[1] http://www.istc.cnr.it/T3/map/index.html

[2] A. Altman and M. Tennenholtz, "Incentive Compatible Ranking Systems", *Proc. of the Sixth International Joint Conference on Autonomous Agents and Multiagent Systems.* (AAMAS'07), 2007.

[3] A. Cheng and E. Friedman, "Sybilproof Reputation Mechanisms", *SIGCOMM'05 Workshops*, Aug 2005, ACM.

[4] D. do B. DeFigueiredo and E. T. Barr, "TrustDavis: A Non-Exploitable Online Reputation System", *In Proceedings of Seventh IEEE International Conference on E-Commerce Technology* (CEC'05), 2005.

[5] E. Friedman and P. Resnick, "The Social Cost of Cheap Pseudonyms", *Journal of Economics & Management Strategy*, vol. 10, n 2, 2001, pp 173–199.

[6] E. Friedman, P. Resnick and R. Sami, "Manipulation-Resistant Reputation Systems", Chap. 27 of Nisan *et. al.* (ed), *Algorithmic Game Theory*, 2007, Cambridge Press, pp 677–697.

[7] D. Gambetta, "Can we Trust Trust?", Chap. 12 of Gambetta (ed) *Trust*, 1990, pp 213–237, Blackwell.

[8] R. Guha, R. Kumar, P. Raghavan and A. Tomkins, "Propagation of Trust and Distrust", in *Proceedings of the Thirteenth International World Wide Web Conference* (WWW2004), 2004, pp 17–22.

[9] S. D. Kamvar, M. T. Schlosser and H. Garcia-Molina, "The EigenTrust Algorithm for Reputation Management in P2P Networks", in *Proceedings of the Twelfth International World Wide Web Conference* (WWW2003), *ACM, 2003,* pp 20–24 .

[10] H. Li and M. Singhal, "Trust Management in Distributed Systems", *IEEE Computer Magazine*, vol 40, issue 2, Feb 2007, pp 45–53.

[11] Z. Malik and A. Bouguettaya, "Reputation Bootstrapping for Trust Establishment among Web Services", *IEEE Internet Computing*, vol 13, number 1, Feb 2009, pp 40–47.

[12] S. P. Marsh, "Formalising Trust as a Computational Concept", *Ph. D. Thesis*, University of Stirling, April 1994.

[13] S. Marti, "Trust and Reputation in Peer-to-Peer Networks", *Ph.D. Thesis*, Stanford University, May 2005.

[14] P. Resnick, R. Zeckhauser, E. Friedman and K. Kuwabara, "Reputation Systems", *Communications of the ACM*, vol. 43, issue 12, Dec 2000, pp 45–48.

[15] T. Riggs and R. Wilensky, "An Algorithm for Automated Rating of Reviewers", in *Proceedings of the First ACM/IEEE-CS joint conference on Digital libraries*, 2001.