# A Game-Theoretic Framework for Analyzing Trust-Inference Protocols

Ruggero Morselli* Jonathan Katz* Bobby Bhattacharjee*

## Abstract

We propose a novel game-theoretic framework for analyzing the robustness of trust-inference protocols in the presence of adversarial (but rational) users. To the best of our knowledge, this is the first such framework which simultaneously (1) admits a rigorous and precise definition, thereby enabling formal proofs of security (in various adversarial settings) for specific trust-inference protocols; (2) is flexible enough to accommodate a full range of (realistic) adversarial behavior and network models; and (3) is appropriate for *decentralized* networks, and in particular does not posit a trusted, centralized party with complete knowledge of the system history. We also show some preliminary results regarding the design of trust-inference protocols which can be rigorously *proven secure* within our model.

In addition to establishing a solid foundation for future work, our framework also enables a rigorous and objective comparison among existing trust inference protocols.

## 1 Introduction

Peer-to-peer networks require a significant amount of cooperation among their members in order to fully realize their potential. For example, in a resource-sharing system where users trade, say, spare computer cycles, such cooperation is crucial to the functioning of the system. Indeed, if too many "free riders" (i.e., those who use system resources without donating any of their own) are present, the utility of the system as a whole — especially for "good" users who *do* donate their resources — will markedly decrease.

Enforcing such cooperation in a peer-to-peer network is, unfortunately, rather difficult, especially because there is no central authority with the ability to "punish" non-cooperating users. To address this issue, a large body of work has recently focused on providing *incentives* for players in the system to behave in a cooperative manner.[1] One straightforward way to attempt to enforce such cooperation is via a "payback" mechanism in which user $A$ donates (some amount of) resources to user $B$ only as long as $B$ continues to donate (a similar amount of) resources to $A$. (I.e., $A$ is willing to extend $B$ only a certain amount of "credit" at which point $B$ must begin paying back.) Such simple mechanisms are, however, rather limited. First, *direct* interaction between $A$ and $B$ may occur infrequently (or even only once!), giving $A$ little or no chance to "redeem" resources donated to $B$. Second, it is unclear what happens if, say, $B$ leaves the system before having had the chance to pay back $A$. A solution of this type also strongly biases users toward interacting *only* with parties with whom they have had direct previous (positive) experience. Although intuitively appealing, this potentially limits the overall utility of the system since users will tend to trade repeatedly with the same partners rather than explore new partners. We note that it also makes it more difficult for new users to be fully integrated into the system. Finally, it is not clear what happens when participants interact for the first time (i.e., when the system is first initialized): when $A$ and $B$ first meet, who first extends credit to whom?

The above drawbacks motivate the idea of having users base their future actions on *more* than just their own personal history of prior interactions; see, e.g., [10] for simulations and discussion further illustrating this point. In particular, one might hope to design *trust-inference*[2] mechanisms by which information about user behavior can be propagated throughout the system; in this way (at least in theory), a user $A$ who freely donates resources will be rewarded as others will be more likely to share resources with

*{ruggero,jkatz,bobby}@cs.umd.edu. Department of Computer Science, University of Maryland. Work supported by NSF Trusted Computing Grant #0310499.

[1] We stress that the intention of such incentives is not to prevent attacks on the system by *malicious* users, but rather to enforce cooperative behavior in the network on the part of *rational* (self-interested) users. Purely malicious behavior must be handled using more traditional security measures.

[2] These are also known in the literature as *trust-propagation protocols* or *reputation/recommender systems*. All of these seek to accomplish essentially the same task.

$A$, while a user $B$ who takes resources without giving any of his own will be punished to the extent that others refuse to share resources with $B$. A growing recognition of the importance of trust inference has led to an extensive amount of research focused on designing "good" trust-inference protocols, both in the specific context of peer-to-peer networks as well as in more general settings. We cannot survey all prior work here, but refer the reader to [6, 13, 17, 19, 20, 12, 1, 3, 2, 10, 14, 5, 18, 9] as representative examples.

Unfortunately, we are aware of very few works which rigorously define what a "good" trust-inference protocol should achieve! Instead, a lot of work in this area has been rather heuristic and ad-hoc, proposing solutions satisfying some list of properties but with no indication that these are the "right" properties one should aim for. In other cases, simulations or informal arguments indicate that a proposed trust-inference protocol is resilient to a *particular* adversarial strategy (or strategies), but no proof is offered to show that the protocol is resilient to *all* (rational) adversarial strategies. Finally, many works make unjustified assumptions; for example, some works assume that although users may cheat by refusing to share resources, all users honestly report the behavior of other nodes (or even their own behavior!).

Some notable exceptions (see [6, 5, 18]) provide a formal adversarial model, a definition of robustness[3], and proofs that a proposed protocol is robust under the given definition. However, all work of this type of which we are aware assumes some form of *global knowledge* which would be implemented in practice using some centralized mechanism. For example, Friedman and Resnick [6] posit that all parties have complete and accurate knowledge of the previous behavior of all other users in the system; other work focusing on the "E-bay model" (see [5]) assumes a public, incorruptible bulletin board on which users post feedback about each other.

### 1.1 Our Contribution

In the course of our ongoing work developing and analyzing protocols for trust inference in completely decentralized systems [11, 16], we have been frustrated by the lack a formal model in which to evaluate our proposed mechanisms, as well as the lack of any objective way to compare the robustness our protocols with previously-proposed ones. The framework we propose here was developed in response to this need, and we hope it will prove useful to other researchers in this area. We stress that the model presented here is very preliminary, but will hopefully serve as a basis and as an impetus for much-needed future work in this domain. (For those who do not agree with the particulars of our model, we hope they agree that *some* formal model is sorely needed!) As we see it, the advantages of our framework include:

- It admits a concrete and precise definition, thereby enabling rigorous proofs of security (in a chosen adversarial model) for specific protocols.
- Similarly, the definition enables an objective way to compare existing trust-inference protocols and to determine their suitability for various systems under a given adversarial model.
- Our definition assumes no "global knowledge", centralized infrastructure, or pre-provisioned trusted parties, and is therefore appropriate for modeling completely decentralized systems with no central authority. However, we note that our model may be easily augmented to include a trusted authority should one choose to do so.
- Our definition is *flexible* enough to allow consideration of a wide range of adversarial behavior and system models (such as adversarial coalitions, Sybil attacks [4], and asynchronous trading) not typically handled (in a formal way) by previous work.

In Section 2 we discuss our framework and define our notion of *robust* trust inference. We also discuss additional desiderata which provide ways of discriminating among robust trust-inference protocols. In Section 3 we give preliminary results indicating that robust protocols are achievable even in very strong adversarial environments (i.e., allowing for arbitrary-size coalitions, Sybil attacks, asynchronous trading, and easy-to-change pseudonyms) *without any centralized infrastructure*. We warn the reader that these results are only meant to illustrate the *feasibility* of realizing our definition of robustness; developing more efficient protocols (which remain provably robust) is the subject of ongoing research.

# 2 Adversarial Framework

We define our adversarial framework in two stages. First, we describe our basic framework which can be used to model essentially any sort of adversarial behavior and/or network in a very simple way. Jumping

[3]Fixing the adversarial model is usually the difficult part, since a robust protocol is almost always defined as one whose actions form a game-theoretic equilibrium (i.e., an adversary has no reason to deviate from the prescribed actions).

ahead a bit, we then define our notion of robustness which will remain unchanged even as the adversarial model is adjusted. (Basically, a trust-inference protocol is *robust* if the actions prescribed by the protocol form a game-theoretic equilibrium. We stress, however, that single-player deviations in our model actually correspond to adversarial *coalitions* in the real network.) Our basic framework gives the adversary a considerable amount of power, and is probably too pessimistic for modeling realistic threats in real-world systems.[4] Thus, we discuss a number of ways of extending our model (which have the effect of restricting the power of the adversary). Our goal here is to highlight the flexibility and generality of our approach, rather than to suggest any particular choice of adversarial model. Indeed, different adversarial models are better suited for different environments and so there is no "best" model to consider.

## 2.1 Basic Framework

A key component of our framework is the notion of a *pseudonym* by which a user is known to others in the network. We assume pseudonyms with the following properties: they are distinct, they are easy to generate by users *themselves* (and do not require the services of a trusted party), and it is impossible to impersonate another party by using their pseudonym. All these properties are (essentially) satisfied by identifying pseudonyms with public keys for a secure digital signature scheme [8].[5] We stress that these public keys are not assumed to be registered in any central location, and need not be certified in any way. In particular, although we assume that honest participants use the same pseudonym throughout their entire lifetime, an adversary can easily generate new pseudonyms as often as it likes.

Our model gives the adversary almost complete control of the system. For convenience, we use the standard conventions of the cryptographic community and model adversarial actions using various *oracles*. Some of these oracles correspond to actions of a real-world adversary, while others merely offer a convenient way of considering the worst-case scenario of events which (in the real world) are outside the adversary's control.[6] Given a trust-inference protocol $\Pi$ (whose details are entirely known to the adversary), we provide adversary $\mathcal{A}$ the following oracles:

- $\mathsf{NewUser}$ creates a new honest user in the system, and $\mathcal{A}$ learns this user's pseudonym. A party using pseudonym $i$ is simply called "user $i$".
- $\mathsf{HonestPlay}(i, j)$ causes honest users $i$ and $j$ to play an instance of some 2-player game (e.g., prisoners' dilemma). In playing this game, the users will behave exactly in accordance with protocol $\Pi$. Note that $\Pi$ prescribes both how trust should be inferred as well as how a user's actions should depend upon the inferred value.
- $\mathsf{Play}(i, id, action)$ plays a 2-player game between $\mathcal{A}$ (using pseudonym $id$) and honest player $i$. The adversary plays *action* while $i$ behaves in accordance with $\Pi$. The adversary may not use an $id$ held by an honest party (this would amount to impersonation, which is assumed impossible).
- $\mathsf{Send}(i, id, msg)$ sends $msg$ to honest player $i$ "from" player $id$, where again we require that $id$ not be held by an honest party. This models messages $\mathcal{A}$ sends as part of $\Pi$ (of course, $\mathcal{A}$ need not behave according to $\Pi$).

We do not provide an oracle enabling honest players to send messages to each other; this is the one part of our model *not* under adversarial control. Instead, we assume that $\Pi$ is executed faithfully among the honest users "in the background" and without any interference from $\mathcal{A}$ (except for messages $\mathcal{A}$ can send on behalf of pseudonyms it controls). We assume that $\mathcal{A}$ can see any messages sent between honest users as part of $\Pi$. (Note that if $\Pi$ is deterministic, then $\mathcal{A}$ automatically knows these messages anyway.)

For simplicity and concreteness, we assume (following [6]) that all 2-party games are a "prisoners' dilemma" with the payouts indicated below (where $C$ represents "cooperate" and $D$ represents "defect"):

| | C | D |
|---|---|---|
| C | (1, 1) | (-1,2) |
| D | (2, -1) | (0, 0) |

In particular, our results in Section 3 assume the payoff matrix above. Note, however, that our framework easily accommodates different games, payouts that change with time (or according to $\Pi$), or adversarial selection of the game to play.

[4]Yet, it is interesting that robust trust-inference protocols exist even for our strongest adversarial model; see Section 3.

[5]We stress two caveats here: first, equating digital signatures with pseudonyms is only sound when considering *computationally-limited* (e.g., poly-time) adversaries, as is typically the case of interest. Second, the implicit assumption is that users will be careful not to "leak" the associated secret key. Maintaining secrecy of secret keys is a security concern that lies outside the game-theoretic framework considered here.

[6]If a protocol is robust even against an ideal adversary having this level of control over the network, then clearly it will also be robust against a real-world adversary.

**Robustness.** In order to model robustness in game-theoretic terms, we need to add a notion of time (as well as a *discount factor*) to our model. Here, we do so in a very general fashion (essentially giving the adversary the most power); we discuss some more restrictive ways of dealing with time below.

We assume that each time the adversary $\mathcal{A}$ makes an oracle call, it associates with the call a particular time $t$ (where $t \geq 0$ is an integer). Other than the fact that the time $t$ can never decrease, our only restriction is that $\mathcal{A}$ "can't do too much in too short a time"; thus, $\mathcal{A}$ can make at most $N$ $\mathsf{NewUser}$ calls with the same value of $t$ (i.e., only some bounded number of users join at any particular time) and at most $N'$ $\mathsf{Play}$ calls with the same value of $t$ (i.e., $\mathcal{A}$ cannot trade with too many people at the same time). Finally, $\Pi$ is assumed to be run whenever the adversary "moves the clock forward". I.e., when $\mathcal{A}$ makes its first oracle call at some time $t$, we assume that the honest players run $\Pi$ immediately beforehand, based on the events that have occurred up to time $t$. We stress that $\mathcal{A}$ may interact with multiple parties at some instant $t$ without giving these parties any chance to run $\Pi$ in the interim.

Note that one may always set $N, N'$ as large as one likes, and thus the above does not fundamentally restrict the adversary's power. However, a given protocol may only be provably secure when $N, N'$ are lower than a certain bound. (The implication is that the protocol is secure against one class of adversaries, but not necessarily secure when the adversary has more power: e.g., in case the adversary releases a virus giving it control over a huge number of hosts.)

We measure the *utility* of the adversary as follows. Each time the adversary makes a $\mathsf{Play}$ oracle call at some time $t$, the adversary's utility increases by $\delta^t \mu$, where $\mu$ is the payoff given by the matrix above (i.e., if $\mathcal{A}$ plays $D$ and the honest user player $C$, then $\mu = 2$) and $\delta < 1$ is a *discount factor* [7]. We assume a rational adversary who wishes to maximize its total utility as time tends to infinity. We may now define what it means for $\Pi$ to be robust:

**Definition 1** $\Pi$ is *robust* if $\mathcal{A}$ maximizes its utility by following $\Pi$; more formally, if the actions prescribed by $\Pi$ form a subgame-perfect equilibrium[7] (cf. [7]).

**Additional desiderata.** We view robustness as a necessary criterion for a trust-inference protocol to satisfy in order to be useful (if $\Pi$ is not robust, than why would *any* party follow $\Pi$?). However, robustness alone is not enough. The following are some additional criteria that must be considered:

- The **expected utility** of $\Pi$ is the utility a participant expects to achieve when everyone is honest. Clearly, higher expected utility is preferable.
- $\Pi$ should ideally be **resilient to trembles** (see [6]) which occur when a player defects or fails to follow $\Pi$ "by mistake", e.g., due to network faults rather than active cheating. The expected utility of $\Pi$ may depend on the probability $\varepsilon$ of trembles, and this should be taken into account.
- A protocol should also be **efficient at admitting new users**. Thus, even though new users may have to "pay their dues" [6, 9], the penalty for newcomers should not be so severe that it discourages users from joining altogether.
- Of course, the **efficiency** of $\Pi$ (in terms of, say, the number of messages that must be sent) is also of interest.

As examples: a protocol that instructs all players to always defect is robust but has expected utility 0. The "grim trigger" strategy [7] (discussed below) is robust and achieves the best possible expected utility when $\varepsilon = 0$; however, it does not perform well when trembles occur with positive probability $\varepsilon > 0$. A protocol in which users do not interact with newcomers as long as reliable "veterans" are available may be robust but does not admit newcomers efficiently.

## 2.2 Extensions

The reader may well notice that the adversarial framework presented above is quite strong, and likely too pessimistic. Yet presenting such a strong framework has a number of advantages: (1) if a trust-inference protocol can be proven robust in such a strong model, it will certainly be robust in real-world adversarial environments; alternately, a "proof" that the model is too strong (in the sense that no reasonable and robust trust-inference protocols exist in that model) would be a very interesting and useful result; (2) the framework is general enough to encompass threats (such as coalitions, Sybil attacks, etc.) not typically modeled by previous work. Furthermore, (3) the framework is flexible enough to allow consideration of a number of more realistic threat models. We discuss some of these briefly now.

**Network membership.** In the model above, we have allowed the adversary to control the size of the network via $\mathsf{NewUser}$ calls. A more realistic model might assume that players continually join at some

[7] Sometimes, we will relax this to require that it only form a Nash equilibrium.

constant rate. The model may further assume that each party leaves the network with some probability $\alpha$ at each time period [6]. Note that using a model which assumes some constant turnover will automatically require a protocol to admit newcomers efficiently if it is to have high expected utility.

**Network interactions.** In the model above, we have allowed the adversary to control the trading partners of the honest parties via HonestPlay queries. While useful insofar as it models the worst-case behavior of the system, this clearly gives the adversary too much control. A more realistic model might have players paired off at random in a given time period. Furthermore, the model might assume that each player interacts exactly once during each time period; this would correspond to a *synchronous* network.

**No coalitions or Sybil attacks.** Often, the simplifying assumption is made that the adversary acts alone (i.e., there are no coalitions) and can only act as a single player would (i.e., the adversary is not powerful enough to simulate the actions of multiple users). In general, we do not view such assumptions as realistic, although we agree that they simplify the analysis. In any case, it is easy to modify our model in the appropriate way (namely, by limiting the adversary to a single Play query per time period) to model this class of adversarial behavior. It is equally easy to modify our model so that a bound on the maximum coalition size is enforced.

# 3 Preliminary Results

We briefly sketch some preliminary results on the design of robust trust-inference protocols. These results reflect work in progress, and are important insofar as they demonstrate what is achievable in the model as sketched above, and also since (to the best of our knowledge) they are the first provably-robust protocols which do not assume any centralized infrastructure.

The first protocol we examine is the "grim trigger" strategy which mandates the following: all players cooperate until the first defection occurs. When defection occurs, the user who interacted with the defector in the previous round informs all players of this fact. Once a user hears that a defection has occurred, that user defects from then on.

**Lemma 1** *The "grim trigger" strategy is robust*[8]*, and achieves optimal expected utility when the probability of trembles is 0, in the strongest adversarial model considered here.*

[8]We note that its actions form a Nash equilibrium, not a subgame-perfect equilibrium.

We present this result only to indicate the feasibility of achieving robust solutions in our model. Of course, one problem with this strategy in practice is that it is not at all resilient to trembles.

Our second protocol is more interesting, and achieves a robust and efficient solution but still without any trusted third party. This protocol is a modification of the "pay-your-dues" protocol of [6]. However, we stress that [6] assume a trusted authority who is also omniscient (and in particular knows the results of all interactions of the previous round), whereas we make no such assumption. Our adversarial model follows [6]: we assume synchronous trading, where in each round players are randomly paired. We also focus on single-player deviations, and assume that coalitions are not a concern. Our protocol $\Pi$ is constructed as follows:

- At the end of each round, each player broadcasts whether its partner from the previous round deviated or complied with the protocol.
- A player $i$ is defined to be a *veteran* if a different player broadcast a message stating that $i$ was compliant in the previous round. All other players are called *newcomers* (note that this category includes both true newcomers as well as any players who deviated).
- In the following round, players trade as follows subject to the exception discussed below (this is exactly as in [6]): if two veterans or two newcomers trade, they both cooperate; if a veteran trades with a newcomer, the veteran defects and the newcomer cooperates (the veteran's defection here is considered to be compliant with the protocol).
- An exception to the above occurs if $i$ is paired with $j$ in the current round, and in the previous round $j$ broadcast a false complaint against $i$. In this case, $i$ defects.

Note that we have essentially replaced the trusted party of [6] with a broadcast stage in which players announce whether their partner of the previous round deviated. However, *we take into account that players may* **lie** *when they broadcast this information* (in [6], the trusted party is assumed both to accurately know what really took place, as well as to reliably inform others of what occurred). In fact, the "exception" (above) is introduced exactly to ensure that lying will not increase the adversary's utility.

**Theorem 1** *(Informal) The above protocol is a robust trust-inference protocol in the adversarial model sketched above, and has positive expected utility even in the presence of trembles.*

# 4 Concluding Remarks

We stress that the framework presented here is a work in progress, and we do not claim that this is the final word on the subject. To the contrary, we hope that this paper inspires further work in this important area; that others will be motivated to refine and augment our model; and that researchers will attempt to design trust-inference protocols which can be rigorously proven to be robust within our framework. We feel strongly that the development and study of formal models for robust trust inference are necessary for future progress in this area.

Our work suggests a number of tantalizing open questions. First, can robust trust-inference protocols with very low communication requirements (in particular, not requiring broadcast) be designed? Alternately, can one show the impossibility of designing very efficient yet robust protocols in a particular adversarial environment? We conjecture that efficient and robust trust inference is impossible (when no trusted authority is assumed) within adversarial models which allow arbitrary-size coalitions/Sybil attacks. It would be wonderful to formalize and rigorously prove (or disprove) this conjecture within the framework given here.

# References


[1] K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In *Proc. Intl. Conf. on Information and Knowledge Management*, 2001.

[2] C. Buragohain, D. Agrawal, and S. Suri. A game theoretic framework for incentives in p2p systems. In *Proc. 3rd Intl. Conf. on Peer-to-Peer Computing*, 2003.

[3] E. Damiani, S.D.C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *9th ACM Conf. on Computer and Communications Security*, 2002.

[4] J.R. Douceur. The sybil attack. In *1st Intl. Workshop on Peer-to-Peer Systems*, 2002.

[5] First interdisciplinary symposium on online reputation mechanisms, Apr 2003. See http://www.si.umich.edu/~presnick/reputation/symposium.

[6] E. Friedman and P. Resnick. The social cost of cheap pseudonyms. *Journal of Economics and Management Strategy*, 10(2):173–199, 1998.

[7] D. Fudenberg and J. Tirole. *Game Theory*. MIT Press, 2002.

[8] S. Goldwasser, S. Micali, and R. Rivest. A digital signature scheme secure against adaptive chosen-message attacks. *SIAM J. Computing*, 17(2):281–308, 1988.

[9] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *Proc. 12th Intl. Conf. on the World Wide Web*, 2003.

[10] K. Lai, M. Feldman, I. Stoica, and J. Chuang. Incentives for cooperation in peer-to-peer networks. In *Workshop on Economics of Peer-to-Peer Systems*, 2003.

[11] S. Lee, R. Sherwood, and B. Bhattacharjee. Cooperative peer groups in NICE. In *IEEE Infocom*, 2003.

[12] R. Lethin. Reputation. In Oram [15], chapter 17.

[13] R. Levien and A. Aiken. Attack-resistant trust metrics for public key certification. In *USENIX Security Symposium*, 1998.

[14] P. Nixon and S. Terzis, editors. *Proc. 1st Intl. Conf. on Trust Mgmt.* Springer-Verlag, 2003.

[15] A. Oram, editor. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*. O'Reilly, 2001.

[16] Project NICE at the University of Maryland. http://www.cs.umd.edu/projects/nice/.

[17] M.K. Reiter and S. Stubblebine. Authentication metric analysis and design. *ACM Trans. Info. and System Security*, 2(2):138–158, 1999.

[18] Reputations research network. See http://databases.si.umich.edu/reputations/.

[19] P. Resnick, K. Kuwabara, B. Zeckhauser, and E. Friedman. Reputation systems. *Comm. ACM*, 43(12):45–48, 2000.

[20] M. Waldman, L.F. Cranor, and A. Rubin. Trust. In Oram [15], chapter 15.