5th International Workshop on Security and Trust Management (STM 2009),
Saint Malo, France, September 2009.

Challenges for Robust Trust and Reputation Systems

Audun Jøsang[1]

UNIK
University of Oslo, Norway

Jennifer Golbeck[2]

Human-Computer Interaction Lab
University of Maryland, USA

Abstract

The purpose of trust and reputation systems is to strengthen the quality of markets and communities by providing an incentive for good behaviour and quality services, and by sanctioning bad behaviour and low quality services. However, trust and reputation systems will only be able to produce this effect when they are sufficiently robust against strategic manipulation or direct attacks. Currently, robustness analysis of TRSs is mostly done through simple simulated scenarios implemented by the TRS designers themselves, and this can not be considered as reliable evidence for how these systems would perform in a realistic environment. In order to set robustness requirements it is important to know how important robustness really is in a particular community or market. This paper discusses research challenges for trust and reputation systems, and proposes a research agenda for developing sound and reliable robustness principles and mechanisms for trust and reputation systems.

Keywords: Trust, reputation, security, robustness, attacks.

1 Introduction

Trust and reputation systems (abbreviated TRS hereafter) represent an important class of decision support tools that can help reduce risk when engaging in transactions and interactions on the Internet. From the individual relying party's viewpoint, a TRS can help reduce the risk associated with any particular interaction. From the service provider's viewpoint, it represents a marketing tool. From the community viewpoint, it represents a mechanism for social moderation and control, as well as a method to improve the quality of online markets and communities.

The same basic principles for creation and propagation of trust and reputation in traditional communities are also used by online TRSs. The main difference is that trust and reputation formation in traditional communities typically is relatively inefficient and relies on physical communication e.g. through word-of-mouth, whereas online TRSs are supported by extremely efficient networks and computer systems. In theory it is possible to

[1] Email: `josang@unik.no`
[2] Email: `jgolbeck@umd.edu`

design very effective trust and reputation management in online communities, but the reliability of computed trust and reputation scores, and thereby the usefulness of the TRS itself, also depends on the robustness of the TRS in question.

Attempts to misrepresent reliability and to manipulate reputation are common in traditional communities. Con artists employ methods to appear trustworthy, e.g. through skillful acting or through the fabrication and presentation of false credentials. Similar types of attacks would also apply to online communities. In case some form of TRS is being used to moderate an online community or market, vulnerabilities in the TRS itself can open up additional attack vectors. It is therefore crucial that TRSs are robust against attacks that could lead to misleading trust and reputation scores. In the worst case, a vulnerable TRS could be turned around and used as an attack tool to maliciously manipulate the computation and dissemination of scores. The consequence of this could be a total loss of community trust caused by the inability to sanction and avoid low quality and deceptive services.

When attacks against a TRS occur it does not normally mean that a server hosting TRS functions is being hacked. Attacks on TRSs typically consist of playing the role of relying parties and/or service entities, and of manipulating the TRS through specific behaviour that is contrary to an assumed faithful behaviour. For example, a relying party that colludes or that is identical to the service entity could provide fake or unfair positive ratings to the TRS with the purpose of inflating the service entity’s score, which in turn would increase the probability of that service entity being selected by other relying parties.

Many other attack scenarios can be imagined that, if successful, would give unfair advantages to the attackers. All such attacks have in common that that they result in the erosion of community trust which in turn would be damaging to services and applications in the affected market or community. TRS robustness can therefore be crucial for the market or community where the TRS is being applied.

This paper focuses on TRS robustness by providing a brief literature overview, by invoking some of the fundamental challenges for creating robust TRSs, and by proposing a set of principles for research in this area.

2 Terminology

Fig.1 illustrates how TRSs are functionally integrated with the interaction partners in a community or market. A service entity is assumed to faithfully provide a service to the relying party. A relying party is assumed to rely on scores for its service selection function, and to faithfully provide ratings about service entities or about specific services to the TRS function.

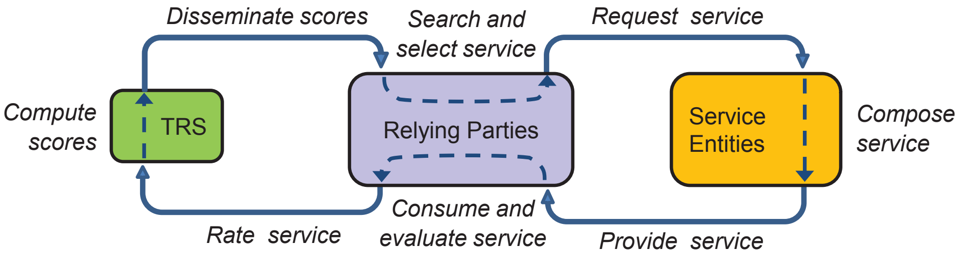

Fig. 1. Functional integration of Trust and Reputation Systems

Note that Fig.1 represents a functional view, not an architectural view. It is for example possible that the TRS function is distributed among all the relying parties as in case of a TRS for P2P networks. It is also possible that there is no distinction between relying parties and service entities.

As Fig.1 indicates, the combination of a TRS and an arbitrary large number of participants (relying parties and service entities) represents a highly dynamic and complex feedback system. Regulating such systems to be robust against malicious manipulation can represent a daunting challenge.

This paper will use two interpretations of trust: *”evaluation trust”* meaning the relying party’s subjective reliability evaluation of a service object or entity, and *”decision trust”* meaning the relying party’s commitment to depend on a service object or entity. Evaluation trust can be formally expressed in terms of probability or belief, whereas decision trust is binary. We will use the term ”reputation” in the sense of the community’s general reliability evaluation of a service entity.

The scope refers to the specific type(s) of trust or reputation assumed by the relying party. In other words, the service or service entity is relied upon to have certain qualities, and the scope is what the relying party assumes those qualities to be. For example, providing financial advice and providing medical advice represent two different scopes for which trust and reputation should be considered separately. Trust and reputation also exist within a context. The context refers to the set of underlying characteristics of the domain in which a TRS is being applied. For example, there can be policies and regulations, and degrees of enforcement of those regulations. Participants will naturally have a higher base trust in each other within a community where law enforcement can be called upon whenever something goes wrong, than they would in a community without law enforcement. The context can influence risk attitudes and how trust and reputation scores are interpreted in practical situations.

In relation to trust systems the term ”recommendation” is often used in the sense of a trust measure passed between entities, whereas the term ”rating” is often used with relation to reputation systems. In this paper we will use the term ”rating” to denote both. A ”score” will refer to a measure of trust or reputation derived by a TRS function based on the received ratings.

3 Challenges for TRS Robustness

3.1 Theoretical Robustness Analysis

The literature on TRSs is relatively well established, where the PhD thesis of Marsh (1994) [21] represents an early study on trust systems and the article by Resnick *et al.* (2000) [23] represents an early introduction to reputation systems. This literature is currently substantial and is still growing fast [7,12]. A large number of TRS designs and architectures have been and continue to be proposed and implemented.

However, the literature specifically focusing on the robustness of TRSs is much more limited and still in an early stage. It should be noted that publications on TRSs usually analyse robustness to a certain extent, but typically only consider a very limited set of attacks. Many of these analyses suffer from the authors’ desire to put their own TRS designs in a positive light, with the result that the robustness analyses often are too superficial and fail to consider realistic attacks. Publications providing comprehensive robustness analyses are rare.

Hoffmann, Zage and Nita-Rotaru (2009) [8] provide a taxonomy and analysis framework for TRSs proposed for P2P networks, and then give an analysis of 24 of the most prominent TRSs based on 25 different attributes. Out of the 24 TRSs, 6 were analysed in more detail because of the representativeness of their characteristics.

The recent paper by Kerr (2009) [17] provides independent robustness analyses of a set of proposed TRSs, and thereby represents a step in the right direction for TRS research. In earlier independent TRS robustness analyses, only a few very prominent commercial TRSs have attracted the attention of independent third party analysts. For example, PageRank has been analysed by Zhang *et al.* (2004) [26] and by Clausen (2004) [3], eBay's Feedback Forum has been analysed by several authors, including Resnick *et al.* (2006) [24].

3.2 Attack Types

This section presents types of threats against TRS that have been described in the literature.

3.2.1 Playbooks

A playbook consists of a sequence of actions that maximises profit or fitness of a participant according to certain criteria. A typical playbook example is to act honestly and provide quality services over a period to gain a high reputation score, and then to subsequently profit from the high reputation score by providing low quality services (at a low production cost). There will be an infinite set of possible playbook sequences, and the actual profit resulting from any particular sequence will be influenced by the actions (and playbooks) of other participants in the community. It should be noted that using playbooks is not necessarily unethical, as explicitly generating oscillation in a brand's reputation is commonly used by commercial players, e.g. by brands of consumer goods. In addition, in numerous natural and artificial agent communities participant agents monitor each others' performance similarly to the way a TRS does it, and try to define the optimal strategy for own survival or to maximize own fitness. This topic is being studied extensively in the economics literature (e.g. [9]) and in the artificial agents literature (e.g. [2,25]). It is only when the norms of a community dictate that specific types of playbooks are unethical that this can be considered as an attack.

3.2.2 Unfair Ratings

This attack consists of providing ratings that do not reflect the genuine opinion of the rater. This behaviour would be considered unethical in most communities and therefore represents an attack. However, it can be extremely difficult to determine when this attack occurs, because agents in a community do not have direct access to each other's genuine opinions, they only see what other agents choose to express. A strategy often proposed in he literature for detecting possible unfair ratings is to compare ratings about the same service entity provided by different agents, and to use ratings from *a priori* trusted agents as a benchmark. However, this method can lead to wrong conclusions in specific scenarios, as e.g. in case of discrimination described below. Situations where it is relatively easy to detect unfair ratings is e.g. in case the quality of the rated or recommended service can be objectively assessed, meaning that the rating can be directly compared to objective quality criteria.

3.2.3 Discrimination

Discrimination means that a service entity provides high quality services to one group of relying parties, and low quality services to another group of relying parties. This behaviour can have very different effects on the service entity's score depending on the specific TRS being used. For example, if a TRS uses a method to detect unfair ratings based on comparing ratings from unknown agents with ratings from *a priori* trusted agents, it is sufficient for the attacker to provide high quality services to the *a priori* trusted agents. The method for detecting unfair ratings will have the paradoxical effect that genuine negative ratings will be rejected by the TRS, so that the attacker will not be sanctioned for providing low quality services.

3.2.4 Collusion

Collusion means that a group of agents coordinate their behaviour which e.g. can consist of running playbooks, of providing unfair recommendations or discrimination. Clever collusion can have significant influence on scores, and thus increase the profit or fitness of certain agents. Collusion is not necessarily unethical, e.g. if it consists of controlling the quality of provided services provided by a group in a coordinated fashion. However, if the collusion consists of coordinating unfair ratings, it would be clearly unethical.

3.2.5 Proliferation

When faced with multiple sellers, a relying party will choose randomly between equal sellers. By offering the same service through many different channels, a single agent will be able to increase the probability of being chosen by a relying party. Proliferation is not necessarily unethical. For example, in some markets it is common that the same product is being marketed by multiple representatives. However, proliferation can be considered unethical in case multiple representations of the same service entity pretend to represent different and independent service entities.

3.2.6 Reputation Lag Exploitation

There is usually a time lag between an instance of a service provision and corresponding rating's effect on the service entity's score. Exploiting this time lag to offer and provide a large number of low quality services over a short period before the rating suffers any significant degradation, would normally be considered unethical.

3.2.7 Re-entry

Re-entry means that an agent with a low score leaves a community and subsequently re-enters the community under a different identity. The effect is that the agent can start from fresh, and thereby avoid the consequences of a low score associated with the previous identity. This would be considered unethical in most situations. Re-entry is a vulnerability caused by weak identity and the inability to detect that two different identities represent the same entity.

An entity can theoretically have multiple identities within the same domain or in different domains. Each identity normally consists of a unique identifier (relative to a domain) and possibly other attributes. There are two cases to consider. In case a reliable mapping is known between an entity's identity in the community and the same entity's identities in other domains (external identities), then it is technically possible to detect re-entry. For example, when the real name of an online user is known in the online community, detection of re-entry will be possible. On the other hand, in case no mapping is known between an entity's identity within the community and external identities, then re-entry will be hard to detect. Even though such mappings may not be publicly known within a community, they may be known by certain parties such as Identity Providers. These parties could assist in detecting re-entry.

3.2.8 Value Imbalance Exploitation

Ratings provided to a TRS typically do not reflect the value of the corresponding transaction. The effect of providing a large number of high quality low value services and a small number of deceptive high value services would then result in a high profit resulting from high value deception without any significant loss in scores. This behaviour is only unethical to the degree that providing deceptive services is unethical in a particular market. If the service entity simply provides low quality high value services, which can not be considered deceptive, then the behaviour could be considered ethical. This threat is related to the problem of mismatch between trust scopes. Weighing ratings as a function of service value is a simple method to remedy this vulnerability.

3.2.9 The Sybil Attack

A single entity that establishes multiple pseudonym identities within a TRS domain is able to provide multiple ratings on the same service object. This represents an attack which can give the attacker an unfair and disproportionately large influence over the computed scores. This attack is called the Sybil attack after a book with the same name by Flora Rheta Schreibe (1973) about a woman suffering from multiple personality disorder.

3.3 Practical Robustness Evaluation

Attack type descriptions such as those provided above only give a theoretical perspective of possible ways that a TRS can be manipulated. It is also important to know how easy it would be to put these attacks into practice.

There is a similarity between evaluating TRS robustness and, for example, evaluating cryptographic strength. The strength of ciphers is typically evaluated against a set of relevant attacks. Major security conferences will for example only accept papers proposing new ciphers on the condition that the designers themselves have evaluated the cipher strength, and it is common for researchers to investigate attacks against each others' cipher designs. This serves the purpose of weeding out week designs, and of raising the standing of researchers who succeed in breaking a cipher. In case of public-key ciphers (used e.g. in SSL and digital signatures) where the theoretical security relies on the difficulty of solving a narrow set of supposedly hard mathematical problems, the estimated strength can roughly be seen as increasing with the time they have been out in the wild without being broken, which is currently about 30 years.

Although it is useful to analyze cipher strength from a purely theoretical perspective, the overall security of any practical cryptographic application such as security protocols and digital signature schemes need to be evaluated in their practical environments. For example, SSL (secure Sockets Layer) which is based on strong cryptography can be considered theoretically secure, but this security protocol is nevertheless vulnerable to phishing attacks due to poor security usability of SSL implementations in web browsers [4,14]. Furthermore, it is relatively simple to attack digital signature schemes in realistic environments, not necessarily due to cryptographic security vulnerabilities, but due to the complexity of the platforms on which they are implemented [16].

Similarly, robustness of TRSs is not simply a theoretical but a practical consideration. It can be useful to analyze TRS robustness from a theoretical perspective, but only when implemented in real environments will it be possible to determine whether a TRS is practically robust. It should also be noted that robustness is relative, e.g. meaning that a TRS design that is robust in one community could be vulnerable in another community. Furthermore, a TRS that is robust at point in time can become vulnerable if the threat picture changes.

Consider for example the Slashdot[3] TRS which has had to evolve in order to provide effective protection against a changing threat picture [12]. The articles posted on Slashdot are selected at the discretion of the Slashdot staff based on submissions from the Slashdot community. Once an article has been posted anyone can comment and rate that article. Shortly after going online in 1997 Slashdot established a team of 25 moderators to deal with rating noise. As the number of Slashdot users and the amount of noise increased, the moderation team grew to 400 moderators. In order to create a more democratic and healthy moderation scheme, automated moderator selection was introduced, and the emerging Slashdot TRS became an integral part of the Slashdot website. The Slashdot TRS actually consists of two layers called M1 and M2, where M1 is for rating comments to articles and M2 is for moderating M1 raters. Slashdot staff are also able to moderate any rating or participant in the Slashdot community, thereby making Slashdot staff omnipotent and able to manually stabilise the system in case Slashdot would be attacked by extreme volumes of noise. This represents a third layer which can be called the control level. Fig2 illustrates the three-layer structure of the Slashdot TRS.

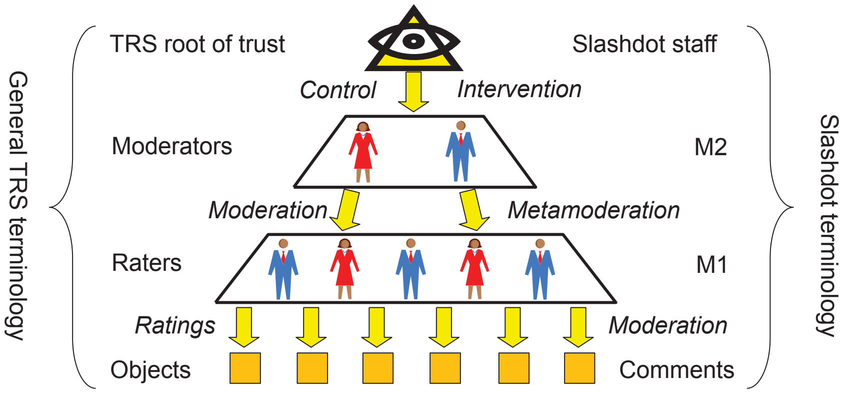

Fig. 2. General model of the Slashdot TRS

It is interesting to note that it probably would be difficult to analyse the robustness of the Slashdot TRS from a purely theoretical perspective due to the large number of variables in the system. The Slashdot reputation system directs and stimulates the massive collaborative effort of moderating thousands of postings every day. The system is constantly being tuned and modified and can be described as an ongoing experiment in search for the best practical

[3] http://slashdot.org/

way to promote quality postings, discourage noise and to make Slashdot as readable and useful as possible for a large community.

The importance of TRS robustness can also be illustrated by events on the Kuro5hin[4] web site which allows members to post articles and comments similarly to Slashdot. Kuro5hin was started in 1999, and the Kuro5hin TRS called *Mojo* underwent major changes in October 2003 because it was unable to effectively counter noise postings from throw-away accounts, and because attackers rated down comments of targeted members in order to make them lose their reputation scores [12]. Some of the changes introduced in Mojo to solve these problems was to only let a comment's score influence a user's Mojo (i.e. reputation score) when there were at least six ratings contributing to it, and to only let one rating count from any single IP address. Kuro5hin survived the attack against its TRS and was able to regain adequate filtering control.

The Advogato TRS [19,20] also has evolved as a reaction to attacks. The Advogato community uses blog and wiki-like features built around open-source software projects. The Advogato TRS is designed to identify trusted members in the community. Users "certify" other people in the system, and the resulting social network and certifications form the foundation for the community-based trust analysis. The Advogato TRS was implemented and tested on the Advogato website which serves as a real-world testbed for the algorithms. In this implementation, the TRS was able to effectively limit access to bad community members. However, simply keeping out bad nodes was not enough to lead to an enjoyable user experience; some correctly-identified good users still posted repetitive or uninteresting content. Addressing this required adding second trust layer using an Eigenvector-based trust metric, similar to PageRank. While a systematic analysis has not been done, anecdotal evidence suggests that the second-layer trust algorithm is effective at filtering interesting posts from noise.

3.4 Importance of Robustness

The examples of Slashdot, Kuro5hin and Advogato illustrate the importance of TRS robustness in online communities. At the same time, TRS are used in systems where their robustness is not always as critical. There are domains where robustness is not claimed or even required for a TRS. For example, when there is little incentive for attacking a TRS, robustness will have little importance. In the case of recommender systems that use trust as a basis for making recommendations and where there is little or nothing to gain from making unfair recommendations [1,6,22], TRS robustness is almost irrelevant.

In order to determine the importance of TRS robustness, it can be useful to consider TRS as an element in a transitive trust chain that in itself also needs to be trusted. In general, the derived trust in a service object based on scores from a TRS can never be stronger than the trust in the TRS itself.

Trust transitivity is based on recommendation between entities, meaning that entities recommend each other in a chained fashion. Thus, there is a difference between trusting an entity to provide a specific service and trusting an entity to recommend somebody who can provide the service [11]. Strictly speaking, trust in the service object is *functional trust* because the service object provides a functional service, whereas trust in the recommending agent is *referral trust* because the recommending agent only refers to another entity (agent

[4] http://www.kuro5hin.org/

or service entity) and is not assumed to provide the functional service. However, there must still be a common shared trust scope for referral and functional trust edges within a specific transitive trust path. The last trust edge of the path must be functional in order to allow derivation of functional trust in the sink service entity or object [11]. Most practical systems, e.g. [6], do not distinguish between functional and referral trust, which is also typical in theoretical designs where referral trust in an agent is directly tied to its functional trustworthiness.

Although strictly speaking a TRS does not trust, scores computed by a TRS can be considered as functional trust in a service object. Similarly, the dissemination of computed scores from a TRS to relying parties can be considered as recommendations. Derivation of functional trust in the service object by the relying party is in principle based on the transitive combination of the relying party's (referral) trust in the TRS and the computed (functional) trust scores provided by the TRS. This is illustrated in Fig3 below.

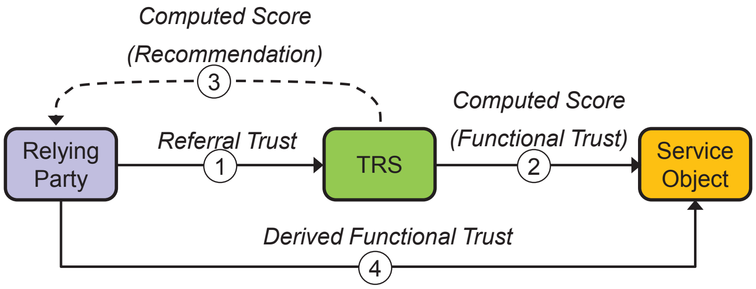

Fig. 3. Initial and derived trust relationships in TRS scenarios

Many different computational models for transitive combination of trust edges have been proposed in the literature. As an example we will use the notation of subjective logic [11,15] where the trust value of e.g. the edge $[\mathrm{RP}, \mathrm{TRS}]$ is denoted as a trust opinion $\omega^{\mathrm{RP}}_{\mathrm{TRS}}$ that can express degrees of uncertainty, and where ”$\otimes$” denotes the computational trust transitivity operator. The relying party's derived trust in the service object is expressed as:

$$\omega^{\mathrm{RP}}_{\mathrm{SO}} = \omega^{\mathrm{RP}}_{\mathrm{TRS}} \otimes \omega^{\mathrm{TRS}}_{\mathrm{SO}} \tag{1}$$

where the attributes RP, TRS and SO denote ”Relying Party”, ”Trust and Reputation System” and ”Service Object” respectively.

The question now arises how the referral trust in the TRS should be determined, and what role TRS robustness plays. We will make the intuitive observation that the correctness of the computed scores will be a function of two main factors: 1) TRS robustness, and 2) efforts by attackers to manipulate scores. The strength of efforts by attackers can be modeled as an increasing function of perceived incentives for attacking the TRS, which e.g. can be estimated by the values at stake, political motives or similar. Referral trust in the TRS can be modeled as an increasing function of the correctness of scores.

Let $\iota \in [0, 1]$ denote the estimated level of incentive for attacking and manipulating the TRS, where $\iota = 0$ represents zero incentive and $\iota = 1$ represents maximum incentive. Let $\rho \in [0, 1]$ denote the level of TRS robustness, where $\rho = 0$ represents total absence of robustness and $\rho = 1$ represents maximum robustness.

Intuitively, a TRS with no robustness against attacks will compute correct scores if there are no incentives to attack it, but it will compute incorrect scores in case of incentives.

Similarly, a very robust TRS will compute mostly correct scores even in the presence of strong incentives to attack it, and will of course produce correct scores in the absence of incentives. A simple *ad hoc* mathematical function that models this intuitive thinking is represented by Eq.(2) which is illustrated in Fig.4. Other mathematical models are also possible.

$$\mathrm{E}(\omega^{\mathrm{RP}}_{\mathrm{TRS}}) = \rho^{\iota}\ , \quad \text{where } \mathrm{E}(\omega) \text{ is the probability expectation value of the opinion } \omega. \qquad (2)$$

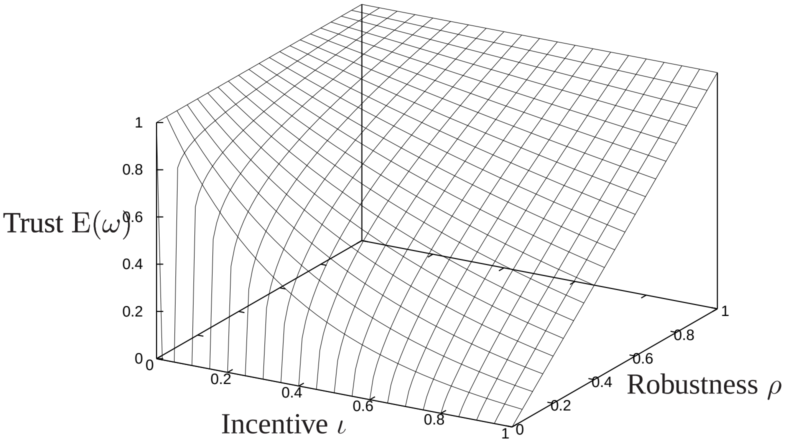

Fig. 4. Evaluation trust in a TRS as a function of attack incentives and of TRS robustness

Deriving a trust opinion $\omega^{\mathrm{RP}}_{\mathrm{TRS}}$ from the scalar $\mathrm{E}(\omega^{\mathrm{RP}}_{\mathrm{TRS}})$ can e.g. be done by uncertainty maximisation [10]. The derived trust value $\omega^{\mathrm{RP}}_{\mathrm{SO}}$ represents referral evaluation trust in the TRS. The decision to rely on the SO is not only a function of the derived evaluation trust, but also of the transaction value at stake and risk attitudes. The higher transaction value and risk aversion, the stronger evaluation trust wold be required to reach decision trust. A model for decision trust as a function of risk and evaluation trust is described in [13].

4 Research Agenda for Designing Robust TRSs

The research literature has so far produced a relatively large body of TRS models. While there is still room for new and innovative TRS designs, the research emphasis should shift more toward designing robustness in TRSs. A prerequisite for robust TRS design is to have reliable methods for evaluating TRS robustness. We see multiple possible approaches that can be applied in isolation or in combination.

- TRS robustness can be evaluated by implementing the TRS in a real environment with potentially malicious and/or unethical participants who have an interest in manipulating the TRS. Different application environments might expose different vulnerabilities in an otherwise identical TRS design. This difference will partly be a function of different incentive structures in different environments.

- TRS robustness evaluations can be conducted from a theoretical perspective by third parties with no interest in the success of the TRS to be evaluated. This would make any robustness evaluation more credible.
- A comprehensive set of robustness evaluation methods and criteria can be defined. This would make it possible for TRS designers to produce credible robustness evaluations of their own TRS designs.

The huge variety in TRS design poses challenges for defining a standardised set of attack types and test methods for evaluating robustness. It therefore seems difficult to define standardised test environments for evaluating all TRS designs. The *”Agent Reputation and Trust Testbed”* (ART) [5] is an example of a TRS testbed that has been specified and implemented by an international group of researchers. However, it is currently not flexible enough for carrying out realistic simulations and robustness evaluations for many of the proposed TRS designs. For example, the study by Kerr (2009) [17] was unable to take advantage of ART for the above mentioned reason, and required designing a totally new test environment in order to carry out the simulations [18].

5 Conclusion

Most TRSs have serious vulnerabilities, and it is questionable whether any TRS can be considered to be robust. Assuming that a TRS would produce unreliable scores in case of attacks, why then would it be used in a particular domain? A possible answer to this question is that in many situations the TRS does not need to be robust because there might be little incentive to attack it, or because the value of the TRS lies elsewhere.

Having a rich interaction surface between participants seems to be an essential factor for growth and consolidation in online communities, and being able to provide feedback through a TRS is in many cases an important interaction dimension. In this perspective, a TRS allows participants to relate more closely to each other, and thereby strengthens the bonding within the community. In that sense, TRS can be seen as a catalyst for growth, in which case robustness is less relevant.

In some cases, there will be a trade-off between TRS robustness and performance, so a lack of robustness should not *per se* be considered a deficiency of any particular TRS. If the authors make claims about robustness, however, testing with accepted methods is important. In either case, we believe that in any TRS proposal or study, at least some consideration of its robustness should be included.

When robustness is seen as a feature of a TRS, presenting the TRS in the context of community accepted norms allows for more accurate and scientific analysis of its benefits and trade-offs. We have identified the value of conducting both theoretical robustness analyses, and evaluating TRS robustness in practical implementations. Theoretical analyses are normally more credible when conducted by independent third parties than by the TRS designers themselves. A common set of attack and analysis methods can be defined and applied in order to make robustness evaluations of own TRS designs more objective. While typical threats should be considered during TRS design and implementation, the evolving threat picture makes it important to ensure that TRS designs are flexible and adaptable to allow the addition of new protection features whenever needed.

In case robustness is important for the overall trust, we have shown that trust in the

TRS can be determined as a function of robustness and of relevant attack incentives. TRS robustness will also be an important consideration when combining multiple TRSs, or when importing scores from one domain into other domains.

References

[1] Paolo Avesani, Paolo Massa, and Roberto Tiella. Moleskiing.it: a trust-aware recommender system for ski mountaineering. *International Journal for Infonomics*, 2005.

[2] Michael Bowling, Brett Browning, and Manuela Veloso. Plays as effective multiagent plans enabling opponent-adaptive play selection. In *In Proceedings of the International Conference on Automated Planning and Scheduling (ICAPS'04)*, 2004.

[3] A. Clausen. The Cost of Attack of PageRank. In *Proceedings of The International Conference on Agents, Web Technologies and Internet Commerce (IAWTIC'2004)*, Gold Coast, July 2004.

[4] Carl Ellison. Ceremony design and analysis, 2007.

[5] Karen K. Fullam et al. A specification of the Agent Reputation and Trust (ART) testbed: experimentation and competition for trust in agent societies. In *Proceedings of the fourth international joint conference on Autonomous agents and multiagent systems (AAMAS'05)*, pages 512–518, 2005.

[6] Jennifer Golbeck. Generating predictive movie recommendations from trust in social networks. In *Proceedings of the Fourth International Conference on Trust Management*, 2006.

[7] Jennifer Golbeck. Trust on the World Wide Web: A Survey. *Foundations and Trends in Web Science*, 1(2), 2008.

[8] Kevin Hoffman, David Zage, and Christina Nita-Rotaru. A Survey of Attack and Defense Techniques for Reputation Systems (to appear). *ACM Computing Surveys*, 42(1), December 2009.

[9] B.A. Huberman and F. Wu. The Dynamics of Reputations. *Computing in Economics and Finance*, 18, 2003.

[10] A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.

[11] A. Jøsang, R. Hayward, and S. Pope. Trust Network Analysis with Subjective Logic. In *Proceedings of the 29th Australasian Computer Science Conference (ACSC2006), CRPIT Volume 48*, Hobart, Australia, January 2006.

[12] A. Jøsang, R. Ismail, and C. Boyd. A Survey of Trust and Reputation Systems for Online Service Provision. *Decision Support Systems*, 43(2):618–644, 2007.

[13] A. Jøsang and S. Lo Presti. Analysing the Relationship Between Risk and Trust. In T. Dimitrakos, editor, *Proceedings of the Second International Conference on Trust Management (iTrust)*, Oxford, March 2004.

[14] A. Jøsang, P.M. Møllerud, and E. Cheung. Web Security: The Emperors New Armour. In *The Proceedings of the European Conference on Information Systems (ECIS2001)*, Bled, Slovenia, June 2001.

[15] A. Jøsang, S. Pope, and S. Marsh. Exploring Different Types of Trust Propagation. In *Proceedings of the 4th International Conference on Trust Management (iTrust)*, Pisa, May 2006.

[16] A. Jøsang, D. Povey, and A. Ho. What You See is Not Always What You Sign. In *Proceedings of the Australian UNIX and Open Systems Users Group Conference (AUUG2002)*, Melbourne, September 2002.

[17] Reid Kerr. Smart Cheaters Do Prosper: Defeating Trust and Reputation Systems. In *Proceedings of the 8th Int. Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, July 2009.

[18] Reid Kerr and Robin Cohen. An Experimental Testbed for Evaluation of Trust and Reputation Systems. In *Proceedings of the Third IFIP International Conference on Trust Management (IFIPTM'09)*, June 2009.

[19] Raph Levien. Attack resistant trust metrics. In Jennifer Golbeck, editor, *Computing with Social Trust*, chapter 5, pages 121–132. Springer, London, UK, 2009.

[20] Raph Levien and Alex Aiken. Attack-resistant trust metrics for public key certification. In *7th USENIX Security Symposium*, pages 229–242, 1998.

[21] S. Marsh. *Formalising Trust as a Computational Concept*. PhD thesis, University of Stirling, 1994.

[22] John O'Donovan and Barry Smyth. Trust in recommender systems. In *IUI'05: Proceedings of the 10th International Conference on Intelligent User Interfaces*, pages 167–174, New York, NY, USA, 2005. ACM.

[23] P. Resnick, R. Zeckhauser, R. Friedman, and K. Kuwabara. Reputation Systems. *Communications of the ACM*, 43(12):45–48, December 2000.

[24] P. Resnick, R. Zeckhauser, J. Swanson, and K. Lockwood. The Value of Reputation on eBay: A Controlled Experiment. *Experimental Economics*, 9(2):79–101, 2006. Avaliable from http://www.si.umich.edu/˜ presnick/papers/postcards/PostcardsFinalPrePub.pdf.

[25] Raquel Ros et al. Retrieving and reusing game plays for robot soccer. In *Proceedings of the 8th European Conference on Case-Based Reasoning (ECCBR-06)*, pages 47–61, Fethiye, Turkey, September 2006.

[26] Hui Zhang et al. Making Eigenvector-Based Reputation Systems Robust to Collusion. In *Proceedings of the Third International Workshop on Algorithms and Models for theWeb-Graph (WAW2004)*, Rome, October 2004.