Trust and Reputation Systems [⋆]

Audun Jøsang

QUT, Brisbane, Australia
a.josang@qut.edu.au
WWW home page http://www.fit.qut.edu.au/∼josang/

Abstract. There are currently very few practical methods for assessing the quality of resources or the reliability of other entities in the online environment. This makes it difficult to make decisions about which resources can be relied upon and which entities it is safe to interact with. Trust and reputation systems are aimed at solving this problem by enabling service consumers to reliably assess the quality of services and the reliability of entities before they decide to use a particular service or to interact with or depend on a given entity. Such systems should also allow serious service providers and online players to correctly represent the reliability of themselves and the quality of their services. In the case of reputation systems, the basic idea is to let parties rate each other, for example after the completion of a transaction, and use the aggregated ratings about a given party to derive its reputation score. In the case of trust systems, the basic idea is to analyse and combine paths and networks of trust relationships in order to derive measures of trustworthiness of specific nodes. Reputation scores and trust measures can assist other parties in deciding whether or not to transact with a given party in the future, and whether it is safe to depend on a given resource or entity. This represents an incentive for good behaviour and for offering reliable resources, which thereby tends to have a positive effect on the quality of online markets and communities. This chapter describes the background, current status and future trend of online trust and reputation systems.

1 Introduction

In the early years of the Internet and the Web, determining whether something or somebody online could be trusted was not thought of as a problem because the Internet community consisted of groups and users motivated by common goals, and with strong trust in each other. The early adopters typically had good intentions because they were motivated by the desire to make the new technology successful. Deceptive and fraudulent behaviour only emerged after the new technology was opened up to the general public and started being used for commercial purposes. The legacy technical architecture and the governance structure of the Internet are clearly inspired by the assumption of well intentioned participants. However, people and organisations currently engaging in Internet activities are not uniformly well intentioned, because they are increasingly

[⋆] Appears in A. Aldini and R. Gorrieri (Eds.), Foundations of Security Analysis and Design IV, FOSAD 2006/2007 Tutorial Lectures. Springer LNCS 4677. ISBN 978-3-540-74809-0. Bertinoro, Italy, September 2007.

motivated by financial profit and personal gain which can lead to unethical and criminal behaviour. The current Internet technology makes us poorly prepared for controlling and sanctioning the substantial and increasing number of users and service providers with unethical, malicious and criminal intentions. As a result, the early optimism associated with the Internet has been replaced by cynicism and diminishing trust in the Internet as a reliable platform for building markets and communities.

As a consequence of this development, the topic of trust in open computer networks is receiving considerable attention in the academic community and the Internet industry. One approach to the problem is to deploy traditional IT security solutions. However, this chapter describes a complementary approach that can be described as *soft security*. The difference between IT security and soft security is explained next.

It is normally assumed that information security technologies, when properly designed, can provide protection against viruses, worms, Trojans, spam email and any other threats that users can be exposed to through the Internet. Unfortunately, traditional IT security technology can only provide protection against some, but not all online security threats. To better understand why, it is useful to look at the definitions of security and of information security separately.

Security can generally be defined as *"the quality or state of being secure - to be free from danger"* [38]. This definition is very broad and covers the protection of life and assets from intentional and unintentional human actions, as well as from natural threats such as storms and earthquakes. In case of protection of information assets, the term information security is normally assumed. Information security is commonly defined as *"the preservation of confidentiality, integrity and availability of Information"* [21], commonly known as the CIA properties. It is here assumed that it is the owner of information assets who has an interest in keeping those assets free from danger, and in preserving their CIA properties. However, in many situations we have to protect ourselves from harmful information assets and from those who offer online resources, so that the problem in fact is reversed. Traditional IT security solutions are totally inadequate for protecting against for example deceitful service providers that provide false or misleading information. We are thus in a situation where we are faced with serious threats, against which there is no established and effective protection. The extended view of online security was first described by Rasmussen & Jansson (1996) [44] who used the term "hard security" for traditional IT security mechanisms like authentication and access control, and "soft security" for what they called social control mechanisms.

In case of traditional IT security, the existence of a security policy is always assumed, whereby the owner of information resources authorises certain parties to perform specific actions. White Hats (i.e. the good guys) and Black Hats (i.e. the bad guys) are easily identified depending on whether they act according to, or against the security policy. In the case of soft security however, this distinction becomes blurred, because there is generally no formally defined or generally accepted policy that defines what constitutes acceptable behaviour. For example, misrepresentation of online services might not even be illegal in the jurisdiction of the service provider, yet a consumer who feels deceived by an online service would most likely define the service provider as a Black Hat. Soft security mechanisms that can provide protection against this type of online threats are typically collaborative and based on input from the whole community. In contrast to traditional IT security where security policies are clearcut and often explicitly defined for a specific security domain by a security manager, soft security is based on an implicit security policy collaboratively emerging from the whole community. On this background we define soft security as follows.

Definition 1 (Soft Security). *Soft security is the collaborative enforcement of, and adherence to common ethical norms by participants in a community.*

While the goal of traditional (hard) information security is to preserve the CIA properties (Confidentiality, Integrity and Availability) of assets within a specific domain, the goal of soft security mechanisms is to stimulate the quality of a specific community in terms of the ethical behaviour and the integrity of its members. What constitutes ethical norms within a community will in general not be precisely defined. Instead it will be dynamically defined by certain key players in conjunction with the average user.

Soft security mechanisms use collaborative methods for assessing the behaviour of members in the community against the ethical norms, making it possible to identify and sanction those participants who breach the norms, and to recognise and reward members who adhere to the norms. A natural side effect is to provide an incentive for good behaviour which in turn has a positive effect on market quality. Reputation systems can be called collaborative praise and sanctioning systems to reflect their collaborative nature. Reputation systems are already being used in successful commercial online applications. There is a rapidly growing literature on the theory and applications of trust and reputation systems. A general observation is that the proposals from the academic community so far lack coherence. The systems being proposed are usually designed from scratch, and only in very few cases are authors building on proposals by other authors.

A survey on trust and reputation systems has been published by Jøsang *et al.* [27]. The purpose of this chapter is to complement that survey and to present the background, current status and the future trend of trust and reputation systems.

Section 2 attempts to define the concepts of trust and reputation, and the objectives of trust management in general. Sections 3 and 4 describe some of the main models and architectures for trust and reputation systems. Sec.5 describes some prominent applications and related issues. The study is rounded off with a discussion in Sec.6.

2 Context and Fundamental Concepts

2.1 The Notion of Trust

Trust is a directional relationship between two parties that can be called *trustor* and *trustee*. One must assume the trustor to be a "thinking entity" in some form meaning that it has the ability to make assessments and decisions based on received information and past experience. The trustee can be anything from a person, organisation or physical entity, to abstract notions such as information or a cryptographic key [22].

A trust relationship has a *scope*, meaning that it applies to a specific purpose or domain of action, such as "being authentic" in the case of a an agent's trust in a cryptographic key, or "providing reliable information" in case of a person's trust in the correctness of an entry in Wikipedia[1]. Mutual trust is when both parties trust each other with the same scope, but this is obviously only possible when both parties are thinking entities. Trust influences the trustor's attitudes and actions, but can also have effects on the trustee and other elements in the environment, for example, by stimulating reciprocal trust [13]. The literature uses the term trust with a variety of meanings [37]. Two main interpretations are to view trust as the perceived reliability of something or somebody, called *"reliability trust"*, and to view trust as a decision to enter into a situation of dependence, called *"decision trust"*.

As the name suggest, reliability trust can be interpreted as the reliability of something or somebody independently of any actual commitment, and the definition by Gambetta (1988) [16] provides an example of how this can be formulated:

Definition 2 (Reliability). Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends.

In Def.2, trust is primarily defined as the trustor's estimate of the trustee's reliability (e.g. expressed as probability) in the context of *dependence* on the trustee.

However, trust can be more complex than Gambetta's definition suggests. For example, Falcone & Castelfranchi (2001) [14] note that having high (reliability) trust in a person is not necessarily sufficient for deciding to enter into a situation of dependence on that person. In [14] they write: *"For example it is possible that the value of the damage per se (in case of failure) is too high to choose a given decision branch, and this independently either from the probability of the failure (even if it is very low) or from the possible payoff (even if it is very high). In other words, that danger might seem to the agent an intolerable risk".*

To illustrate the difference between reliability trust and decision trust with a practical example, consider a fire drill where participants are asked to abseil from the third floor window of a house using a rope that looks old and appears to be in a state of deterioration. In this situation, the participants would assess the probability that the rope will hold him while abseiling. A person who thinks that the rope could rupture would distrust the rope and refuse to use it. This is illustrated on the left side in Fig.1.

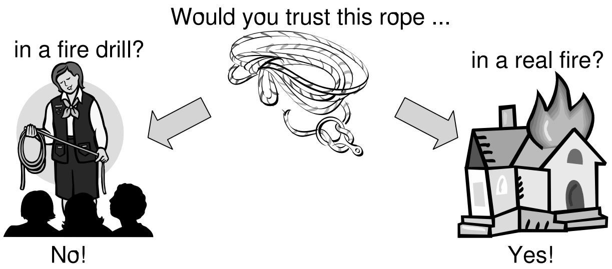


Fig. 1. Same reliability trust, but different decision trust

[1] http://www.wikipedia.org/

Imagine now that the same person is trapped in a real fire, and that the only escape is to descend from the third floor window using the same old rope. In this situation it is likely that the person would trust the rope, even if he thinks it is possible that it could rupture. This change in trust decision is perfectly rational because the likelihood of injury or death while abseiling is assessed against the likelihood of smoke suffocation and death by fire. Although the *reliability trust* in the rope is the same in both situations, the *decision trust* changes as a function of the comparatively different utility values associated with the different courses of action in the two situations. The following definition captures the concept of decision trust.

Definition 3 (Decision). Trust is the extent to which a given party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible.

In Def.3, trust is primarily defined as the willingness to rely on a given object, and specifically includes the notions of *dependence* on the trustee, and its *reliability*. In addition, Def.3 implicitly also covers situational elements such as *utility* (of possible outcomes), *environmental factors* (law enforcement, contracts, security mechanisms etc.) and *risk attitude* (risk taking, risk averse, etc.).

Both reliability trust and decision trust reflect a positive belief about something on which trustor depends for his welfare. Reliability trust is most naturally measured as a discrete or continuous degree of reliability, whereas decision trust is most naturally measured in terms of a binary decision. While most trust and reputation models assume reliability trust, decision trust can also modelled. Systems based on decision trust models should be considered as decision making tools.

The difficulty of capturing the notion of trust in formal models in a meaningful way has led some economists to reject it as a computational concept. The strongest expression for this view has been given by Williamson (1993) [52] who argues that the notion of trust should be avoided when modelling economic interactions, because it adds nothing new, and that well studied notions such as reliability, utility and risk are adequate and sufficient for that purpose. Personal trust is the only type of trust that can be meaningful for describing interactions, according to Williamson. He argues that personal trust applies to emotional and personal interactions such as love relationships where mutual performance is not always monitored and where failures are forgiven rather than sanctioned. In that sense, traditional computational models would be inadequate e.g. because of insufficient data and inadequate sanctioning, but also because it would be detrimental to the relationships if the involved parties were to take a computational approach. Non-computation models for trust can be meaningful for studying such relationships according to Williamson, but developing such models should be done within the domains of sociology and psychology, rather than in economy.

In the light of Williamson's view on modelling trust it becomes important to judge the purpose and merit of trust management itself. Can trust management add anything new and valuable to the Internet technology and economy? The answer, in our opinion, is definitely yes. The value of trust management lies in the architectures and mechanisms for collecting trust relevant information, for efficient, reliable and secure processing, for distribution of derived trust and reputation scores, and for taking this information into account when navigating the Internet and making decisions about online activities and transactions. Economic models for risk taking and decision making are abstract and do not address how to build trust networks and reputation systems. Trust management specifically addresses how to build such systems, and can in addition include aspects of economic modelling whenever relevant and useful.

It can be noted that the traditional cues of trust and reputation that we are used to observe and depend on in the physical world are missing in online environments. Electronic substitutes are therefore needed when designing online trust and reputation systems. Furthermore, communicating and sharing information related to trust and reputation is relatively difficult, and normally constrained to local communities in the physical world, whereas IT systems combined with the Internet can be leveraged to design extremely efficient systems for exchanging and collecting such information on a global scale. Motivated by these basic observations, the design of trust and reputation systems should focus on:

a. Finding adequate online substitutes for the traditional cues to trust and reputation that we are used to in the physical world, and identifying new information elements (specific to a particular online application) which are suitable for deriving measures of trust and reputation.
b. Taking advantage of IT and the Internet to create efficient systems for collecting that information, and for deriving measures of trust and reputation, in order to support decision making and to improve the quality of online markets.

These simple principles invite rigorous research in order to answer some fundamental questions: What information elements are most suitable for deriving measures of trust and reputation in a given application? How can these information elements be captured and collected? What are the best principles for designing such systems from a theoretic and from a usability point of view? Can they be made resistant to attacks of manipulation by strategic agents? How should users include the information provided by such systems into their decision process? What role can these systems play in the business model of commercial companies? Do these systems truly improve the quality of online trade and interactions? These are important questions that need good answers and corresponding solutions in order for trust and reputation systems to reach their full potential in online environments.

2.2 Reputation and Trust

The concept of reputation is closely linked to that of trustworthiness, but it is evident that there is a clear and important difference. For the purpose of this study, we will define reputation according to Merriam-Webster's online dictionary [38].

Definition 4 (Reputation). *The overall quality or character as seen or judged by people in general.*

This definition corresponds well with the view of social network researchers [15, 36] that reputation is a quantity derived from the underlying social network which is globally visible to all members of the network. The difference between trust and reputation can be illustrated by the following perfectly normal and plausible statements:

a. *"I trust you because of your good reputation."*
b. *"I trust you despite your bad reputation."*

Assuming that the two sentences relate to the same trust scope, statement a) reflects that the relying party is aware of the trustee's reputation, and bases his trust on that. Statement b) reflects that the relying party has some private knowledge about the trustee, e.g. through direct experience or intimate relationship, and that these factors overrule any (negative) reputation that a person might have. This observation reflects that trust ultimately is a personal and subjective phenomenon that that is based on various factors or evidence, and that some of those carry more weight than others. Personal experience typically carries more weight than second hand trust referrals or reputation, but in the absence of personal experience, trust often has to be based on referrals from others.

Reputation can be considered as a collective measure of trustworthiness (in the sense of reliability) based on the referrals or ratings from members in a community. An individual's subjective trust can be derived from a combination of received referrals and personal experience. In order to avoid dependence and loops it is required that referrals be based on first hand experience only, and not on other referrals. As a consequence, an individual should only give subjective trust referral when it is based on first hand evidence or when second hand input has been removed from its derivation base [30]. It is possible to abandon this principle for example when the weight of the trust referral is normalised or divided by the total number of referrals given by a single entity, and the latter principle is e.g. applied in Google's PageRank algorithm [43] described in more detail in Sec.5.2 below.

Reputation can relate to a group or to an individual. A group's reputation can for example be modelled as the average of all its members' individual reputations, or as the average of how the group is perceived as a whole by external parties. Tadelis' (2001) [51] study shows that an individual belonging to to a given group will inherit an *a priori* reputation based on that group's reputation. If the group is reputable all its individual members will *a priori* be perceived as reputable and vice versa.

2.3 Trust Transitivity

Trust transitivity means, for example, that if Alice trusts Bob who trusts Eric, then Alice will also trust Eric. This assumes that Bob actually tells Alice that he trusts Eric, which is called a *recommendation*. This is illustrated in Fig.2, where the indexes indicate the order in which the trust relationships and recommendations are formed.

Trust is only conditionally transitive [8]. For example the fact that Alice trusts Bob to look after her child, and Bob trusts Eric to fix his car, does not imply that Alice trusts Eric for looking after her child, nor for fixing her car. However, under certain semantic constraints [30], trust can be transitive, and a trust system can be used to derive trust. In the last example, trust transitivity collapses because the scopes of Alice's and Bob's trust are different.

Based on the situation of Fig.2, let us assume that Alice needs to have her car serviced, so she asks Bob for his advice about where to find a good car mechanic in town. Bob is thus trusted by Alice to know about a good car mechanic and to tell his honest opinion about that. Bob in turn trusts Eric to be a good car mechanic.

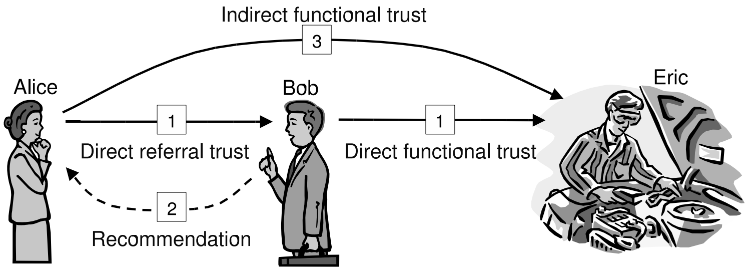


Fig. 2. Transitive trust principle

It is important to separate between trust in the ability to recommend a good car mechanic which represents *referral trust*, and trust in actually being a good car mechanic which represents *functional trust*. The scope of the trust is nevertheless the same, namely to be a good car mechanic. Assuming that, on several occasions, Bob has proven to Alice that he is knowledgeable in matters relating to car maintenance, Alice's referral trust in Bob for the purpose of recommending a good car mechanic can be considered to be *direct*. Assuming that Eric on several occasions has proven to Bob that he is a good mechanic, Bob's functional trust in Eric can also be considered to be direct. Thanks to Bob's advice, Alice also trusts Eric to actually be a good mechanic. However, this functional trust must be considered to be *indirect*, because Alice has not directly observed or experienced Eric's skills in servicing and repairing cars.

Let us slightly extend the example, wherein Bob does not actually know any car mechanics himself, but he trusts Claire, whom he believes knows a good car mechanic. As it happens, Claire is happy to recommend the car mechanic named Eric. As a result of transitivity, Alice is able to derive trust in Eric, as illustrated in Fig.3, where dr-trust denotes direct referral trust, df-trust denotes direct functional trust, and if-trust denotes indirect functional trust.

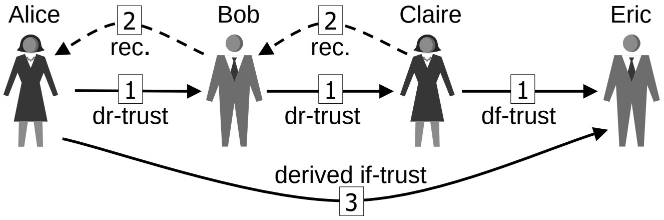


Fig. 3. Trust derived through transitivity

Defining the exact scope of Alice's trust in Bob is more complicated in this extended example. It seems that Alice trusts Bob to recommend somebody (who can recommend somebody etc.) who can recommend a good car mechanic. The problem with this type of formulation is that the length of the trust scope expression becomes proportional with the length of the transitive path, so that the trust scope expression rapidly becomes unmanageable. It can be observed that this type of trust scope has a recursive structure that can be exploited to define a more compact expression for the trust scope. As already mentioned, trust in the ability to recommend represents referral trust, and is precisely what allows trust to become transitive. At the same time, referral trust always assumes the existence of a functional trust scope at the end of the transitive path, which in this example is about being a good car mechanic.

The "referral" variant of a trust scope can be considered to be recursive, so that any transitive trust chain, with arbitrary length, can be expressed using only one trust scope with two variants. This principle is captured by the following criterion.

Definition 5 (Functional Trust Derivation Criterion). *Derivation of functional trust through referral trust, requires that the last trust arc represents functional trust, and all previous trust arcs represents referral trust.*

In practical situations, a trust scope can be characterised by being general or specific. For example, knowing how to change wheels on a car is more specific than to be a good car mechanic, where the former scope is a subset of the latter. Whenever the functional trust scope is equal to, or a subset of the referral trust scopes, it is possible to form transitive paths. This can be expressed with the following consistency criterion.

Definition 6 (Trust Scope Consistency Criterion). *A valid transitive trust path requires that the trust scope of the functional/last arc in the path be a subset of all previous arcs in the path.*

Trivially, every arc can have the same trust scope. Transitive trust propagation is thus possible with two variants (i.e. functional and referral) of a single trust scope.

A transitive trust path stops at the first functional trust arc encountered. It is, of course, possible for a principal to have both functional and referral trust in another principal, but that should be expressed as two separate trust arcs. The existence of both a functional and a referral trust arc, e.g. from Claire to Eric, should be interpreted as Claire having trust in Eric not only to be a good car mechanic, but also to recommend other car mechanics.

The examples above assume some sort of absolute trust between the agents in the transitive chain. In reality trust is never absolute, and many researchers have proposed to express trust as discrete verbal statements, as probabilities or other continuous measures. When applying computation to such trust measures, intuition dictates that trust should be weakened or diluted through transitivity. Revisiting the above example, this means that Alice's derived trust in the car mechanic Eric through the recommenders Bob and Claire can be at most as strong or confident as Claire's trust in Eric. How trust strength and confidence should be formally represented depends on the particular formalism used.

It could be argued that negative trust in a transitive chain can have the paradoxical effect of strengthening the derived trust. Take for example the case where Alice distrusts Bob, and Bob distrusts Eric. In this situation, it might be reasonable for Alice to derive positive trust in Eric, since she thinks “Bob is trying to trick me, I will not rely on him”. When using the principle that the enemy of my enemy is my friend, the fact that Bob recommends distrust in Eric should count as a pro-Eric argument from Alice’s perspective. The question of how transitivity of distrust should be interpreted can quickly become very complex because it can involve multiple levels of deception. Models based on this type of reasoning have received minimal attention in the trust and reputation systems literature, and it might be argued that the study of such models belongs to the intelligence analysis discipline, rather than online trust management. However, the fundamental issues and problems are the same in both disciplines

The analysis of transitive trust relating to the example of Fig.3 uses a rich set of semantic elements. In practical systems and implementations it might be necessary to use simplified models, e.g. by not making any distinction between referral and functional trust, or between direct and indirect trust, and by not specifying trust scopes. This is because it might not be possible to obtain detailed information for making distinctions between trust semantics, and because it would require overly complex mathematical models to take the rich set of aspects into account.

2.4 IT Security and Trust

The term trust is being used extensively in the context if IT security where it can take various meanings. The concepts of Trusted Systems and TCB (Trusted Computing Base) are among the earliest examples of this (see e.g. Abrams 1995 [3]). A trusted system can simply be interpreted as a system designed with strong security as a major goal, and the TCB as the set of hardware and software components that contribute to the security. The concept of evaluation assurance level is a standardised measure of security for trusted systems[2]. Some organisations require systems with high assurance levels for high risk applications or for processing sensitive information. In an informal sense, the assurance level expresses a level of (reliability) trustworthiness of given system. However, it is evident that additional information, such as warnings about newly discovered security flaws, can carry more weight than the evaluation assurance level when users form their own subjective opinion about a trusted system.

More recently, the concept of TC (Trusted Computing) has been introduced by the industry. In general, TC can be defined as information processing on a platform with specialised security hardware. More specifically, TC can mean information processing on a platform equipped with a TPM (Trusted Platform Module) hardware chip that provides specific functionality as standardised by the TCG (Trusted Computing Group) [3].

The term Trust Management has been, and still is used with the relatively narrow meaning of distributed access control, which was in fact the first usage of the term [5]. According to this interpretation, the owner of a resource can determine whether a third party can be trusted to access resources based on attribute certificates that can be chained in a transitive fashion. The related concept of Trust Negotiation is used to

[2] See e.g. the UK CESG at http://www.cesg.gov.uk/ or the Common Criteria Project at http://www.commoncriteriaportal.org/

[3] https://www.trustedcomputinggroup.org/home

describe the process of exchanging access credentials and certificates between a requestor and the resource owner with the purpose of determining whether the requestor is authorised to access the resources.

In identity management, the term Circle of Trust is defined by the Liberty Alliance[4] to denote a group of organisations that have entered into an agreement of mutual acceptance of security and authentication assertions for authentication and access control of users. The Liberty alliance has adopted SAML2.0 [42] as the standard for specifying such security assertions. The WS-Trust standard [5] which has been developed mainly by IBM and Microsoft specifies how to define security assertions that can be exchanged with the WS-Security protocol. WS-Trust and WS-Security have the same purpose as, but are incompatible with SAML2.0. It remains to be seen which of these standards will survive in the long run. Other trust related IT terms are for example

- TTP (Trusted Third Party), which normally denotes an entity that can keep secrets
- Trusted Code, which means a program that runs with system or root privileges
- Trust Provider, which can mean a CA (Certificate Authority) in a PKI.

In cryptography and security protocol design, trust is often used to denote the beliefs in the initial assumptions and in the derived conclusions. In that sense, security protocols represent mechanisms for propagating trust from where it exists (i.e. the initial assumptions) to where it is needed (i.e. the conclusions). Analysing this form of trust propagation can be done with formal logics and formal methods [50].

The meanings of the various trust related terms used by the IT security community can in general not be intuitively derived and understood solely from the terms themselves. Instead they often have a complex meaning that must be explained in order to be properly understood. The purpose of using trust related terms is twofold: they provide a short and practical metaphor for something that would be tedious to explain each time, and they can also represent marketing slogans to promote particular solutions or interests. The TPM is for example criticised for representing DRM (Digital Rights Management) technology that creates unnecessary complexity and increased cost in PCs and media devices and that can be used to lock users to specific vendors[6]. Applying the term "trusted computing" to this technology has the deceptive marketing effect of defusing public criticism because it sounds like it protects the users whereas in reality it does not[7].

In general, security mechanisms protect systems and data from being adversely affected by malicious and non-authorised parties. The effect of this is that those systems and data can be considered more reliable, and thus more trustworthy. A side effect of implementing strong security is that the functionality and flexibility suffer, so that there is a trade-off between security on the one hand and functionality/flexibility on the other. It is therefore clear that the potential for business applications can suffer with increased real security. On the other hand, users and organisations will tend to use systems that

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

[5] http://www.ibm.com/developerworks/library/specification/ws-trust/

[6] See e.g. the TC FAQ at http://www.cl.cam.ac.uk/∼rja14/tcpa-faq.html and the Content Protection Cost Analysis at http://www.cs.auckland.ac.nz/∼pgut001/pubs/vista_cost.html

[7] See e.g. the animation about TC at http://www.lafkon.net/tc/

they trust, and increased perceived security is a contributing factor for increased trust, which in turn is a catalyst for the uptake of online activity and business. Because real and perceived security seems to have opposite effects on e-business it is interesting to look at their combined effect, as illustrated in Fig.4.

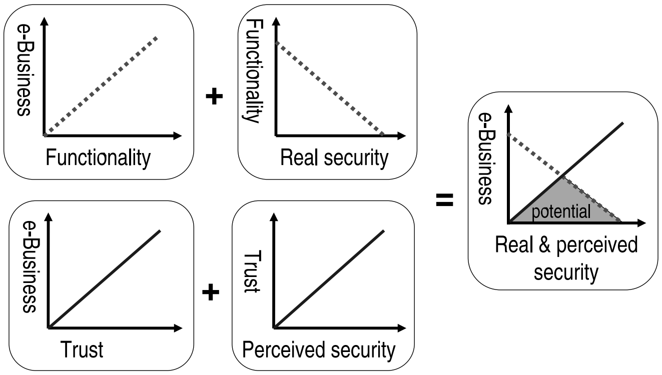


Fig. 4. Combining the effects of real and perceived security on e-business

The shaded triangle on the right hand side graph represents the potential for e-business which is bounded by the effect of decreased functionality as a result of real security, and by the effect of distrust as a result of perceived insecurity. Assuming that the levels of perceived and real security are equal, the optimal situation would be to have a moderate level of real security which results in a moderate level of trust. If it were possible to separate perceived security from real security, it could be optimal to decrease the level of real security and artificially boost the level of perceived security. There is evidence of this happening in the e-commerce industry. For example, online banks used to instruct their customers to look for the locked padlock in the corner of the Web browser as an indicator for transaction security in the form of SSL-based encryption and authentication. That was until phishing attacks emerged. In contradiction to what was claimed, SSL does not provide any practical authentication. The padlock gives a false impression of security because Web browsers display it even when connected to phishing Websites. However, the padlock initially had the effect of making customers trust and start using the Internet as a medium for conducting bank transactions. With the realisation that SSL does not provide any practical authentication, perceived Web security has subsequently been adjusted to correspond better with real Web security.

2.5 Collaborative filtering and Collaborative Sanctioning

Collaborative filtering systems (CF) have similarities with reputation systems in the sense that both types of systems collect ratings from members in a community. However they also have fundamental differences. The assumptions behind CF systems is that different people have different tastes, and rate things differently according to subjective taste. If two users rate a set of items similarly, they share similar tastes, and are grouped in the same cluster. This information can be used to recommend items that one participant likes, to other members of the same cluster. Implementations of this technique represent a form of *recommender systems* which is commonly used for targeted marketing. This must not be confused with reputation systems which are based on the seemingly opposite assumption, namely that all members in a community should judge a product or service consistently. In this sense the term *"collaborative sanctioning"* (CS) [39] has been used to describe reputation systems, because the purpose is to sanction poor service providers, with the aim of giving an incentive for them to provide quality services.

CF takes ratings subject to taste as input, whereas reputation systems take ratings assumed insensitive to taste as input. People will for example judge data files containing film and music differently depending on their taste, but all users will judge files containing viruses to be bad. CF systems can be used to select the preferred files in the former case, and reputation systems can be used to avoid the bad files in the latter case. There will of course be cases where CF systems identify items that are invariant to taste, which simply indicates low usefulness of that result for recommendation purposes. Inversely, there will be cases where ratings that are subject to personal taste are being fed into reputation systems. The latter can cause problems, because a reputation system would normally interpret difference in taste as difference in service provider reliability, potentially leading to misleading reputation scores.

There is a great potential for combining CF and reputation systems, e.g. by filtering reputation scores to reflect ratings from users with a common taste. This could result in more reliable reputation scores. Theoretic schemes include Damiani *et al.*'s (2002) proposal to separate between provider reputation and resource reputation in P2P networks [11].

3 Trust Models and Systems

The main differences between trust and reputation systems can be described as follows: Trust systems produce a score that reflects the relying party's subjective view of an entity's trustworthiness, whereas reputation systems produce an entity's (public) reputation score as seen by the whole community. Secondly, transitivity of trust paths and networks is an explicit component in trust systems, whereas reputation systems usually do not take transitivity into account, or only in an implicit way. Finally, trust systems take subjective expressions of (reliability) trust about other entities as input, whereas reputation systems take ratings about specific (and objective) events as input.

There can of course be trust systems that incorporate elements of reputation systems and vice versa, so that it is not always clear how a given systems should be classified. The descriptions of the various trust and reputation systems below must therefor be seen in this light.

Interpreting trust or trustworthiness as a measure of reliability allows a whole range of metrics to be applied, from discrete to continuous and normalised metrics. This section gives a brief overview of these approaches.

3.1 Discrete Trust Models

Humans are often better able to rate performance in the form of discrete verbal statements, than in the form of continuous measures. A system that allows trust to be expressed in the form of a discrete statement like *"usually trusted"* provides better usability than in the form of a probability value. This is because the meaning of discrete verbal statements comes to mind immediately, whereas probability values require more cognitive effort to be interpreted. Some systems, including [1, 6, 7, 35, 55] are based on discrete trust models.

Discrete measures do not easily lend themselves to sound computational principles. Instead, heuristic methods such as look-up tables must be used. The software encryption tool PGP uses discrete measures for expressing and analysing trust in public keys. PGP implements a very pragmatic approach to the complex issue of deriving trust from a trust network, and is described in more detail in Sec.5.1.

3.2 Probabilistic Trust Models

The advantage of probabilistic models is that the rich body of probabilistic methods can be directly applied. This provides a great variety of possible derivation methods, from simple models based on probability calculus to models using advanced statistical methods. An overview of Bayesian approaches is provided in [32].

Certain models require normalisation in order to produce consistent results. This is for example the case for Google's PageRank algorithm [43]. This is because PageRank requires additivity (i.e. that the sum of probabilities equals one) over the whole population of Web pages. This means that a Web page can only increase its rank at the cost of others. PageRank can also be described as a flow models because it computes trust or reputation by transitive iteration through looped or arbitrarily long chains. PageRank is described in more detail in Sec.8

Other flow models are the Appleseed algorithm [54], Advogato's reputation scheme [33], and the EigenTrust model [31]. The latter computes agent trust scores in P2P networks through repeated and iterative multiplication and aggregation of trust scores along transitive chains until the trust scores for all agent members of the P2P community converge to stable values.

3.3 Belief Models

Belief theory is a framework related to probability theory, but where the sum of probabilities over all possible outcomes not necessarily add up to 1, and the remaining probability is interpreted as uncertainty.

Jøsang (1999,2001) [23, 24] has proposed a belief/trust metric called *opinion* denoted by $\omega_x^A = (b, d, u, a)$, which expresses the relying party A's belief in the truth of statement x. Here b, d, and u represent belief, disbelief and uncertainty respectively where $b, d, u \in [0, 1]$ and $b + d + u = 1$. The parameter $a \in [0, 1]$ represents the base rate in the absence of evidence, and is used for computing an opinion's probability expectation value $E(\omega_x^A) = b + au$, meaning that a determines how uncertainty shall contribute to $E(\omega_x^A)$. When the statement x for example says *"David is honest* *and reliable"*, then the opinion can be interpreted as reliability trust in David. As an example, let us assume that Alice needs to get her car serviced, and that she asks Bob to recommend a good car mechanic. When Bob recommends David, Alice would like to get a second opinion, so she asks Claire for her opinion about David. This situation is illustrated in fig. 5 below.

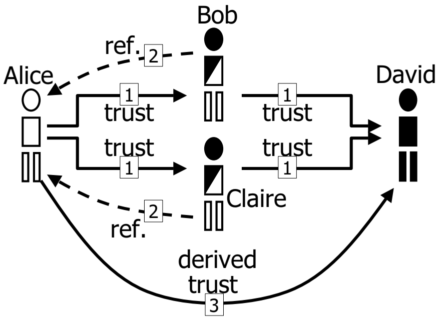


Fig. 5. Deriving trust from parallel transitive chains

When trust and trust referrals are expressed as opinions, each transitive trust path Alice→Bob→David, and Alice→Claire→David can be computed with the *discounting operator*, where the idea is that the referrals from Bob and Claire are discounted as a function Alice's trust in Bob and Claire respectively. Finally the two paths can be combined using the cumulative *consensus operator* or by the averaging operator. These operators form part of *Subjective Logic* [24, 25], and semantic constraints must be satisfied in order for the transitive trust derivation to be meaningful [30]. Opinions can be uniquely mapped to beta PDFs, and this sense the consensus operator is equivalent to the Bayesian updating described in Sec.4.3. This model is thus both belief-based and Bayesian.

By assuming Alice's trust in Bob and Bob's trust in Claire to be positive but not absolute, Alice's derived trust in Eric is intuitively weaker than Claire's trust in Eric.

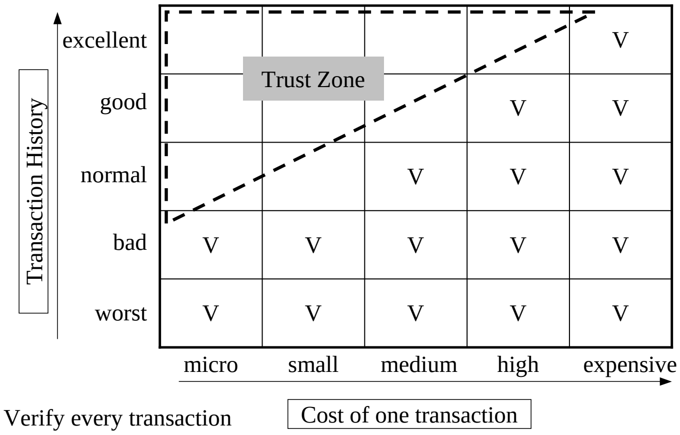
Claire obviously recommends to Bob her opinion about Eric as a car mechanic, but Bob's recommendation to Alice is ambiguous. It can either be that Bob passes Claire's recommendation unaltered on to Alice, or that Bob derives indirect trust in Eric which he recommends to Alice. The latter way of passing recommendations can create problems, and it is better when Alice receives Claire's recommendation unaltered.

3.4 Fuzzy Models

Trust and reputation can be represented as linguistically fuzzy concepts, where membership functions describe to what degree an agent can be described as e.g. trustworthy or not trustworthy. Fuzzy logic provides rules for reasoning with fuzzy measures of this type. The scheme proposed by Manchala (1988) [35] described in Sec.2 as well as the REGRET reputation system proposed by Sabater & Sierra (2001,2002) [46–48] fall in this category. In Sabater & Sierra's scheme, what they call *individual reputation* is derived from private information about a given agent, what they call *social reputation* is derived from public information about an agent, and what they call *context dependent reputation* is derived from contextual information.

3.5 Modelling Decision Trust

There are only a few computational trust models that explicitly take risk into account [17]. Studies that combine risk and trust include Manchala (1998) [35] and Jøsang & Lo Presti (2004) [29]. The system described by Manchala (1998) [35] avoids expressing measures of trust directly, and instead develops a model based on trust-related variables such as the cost of the transaction and its history, and defines risk-trust decision matrices as illustrated in Figure 6. The risk-trust matrices are then used together with fuzzy logic inference rules to determine whether or not to transact with a particular party.



	micro	small	medium	high	expensive
excellent					V
good				V	V
normal			V	V	V
bad	V	V	V	V	V
worst	V	V	V	V	V

Fig. 6. Risk-trust matrix (from Manchala (1998) [35]).

Manchala's risk-trust matrix is intuitive and simple to apply. The higher the value at stake, the more positive experiences are required to decide to trust.

Jøsang and Lo Presti use simple economic modelling, taking into account probability of success, gain, risk attitude and asset value at stake. Let F_C express the fraction of capital at stake, meaning that the relying party is investing fraction F_C of its total capital in the transaction. Let G_s express the gain factor and let p express the probability of success of the transaction. Intuitively F_C increases with G_s when p is fixed, and similarly F_C increases with p when G_s fixed. In order to illustrate this general behaviour let a given agent's risk attitude for example be determined by the function:

$$F_C(p, G_\mathrm{s}) = p^{\frac{\lambda}{G_\mathrm{s}}} \qquad (1)$$

where $\lambda \in [1, \infty]$ is a factor moderating the influence of the transaction gain G_s on the fraction of total capital that the relying party is willing to put at risk. The term *decision surface* describes the type of surface illustrated in Figure 7.

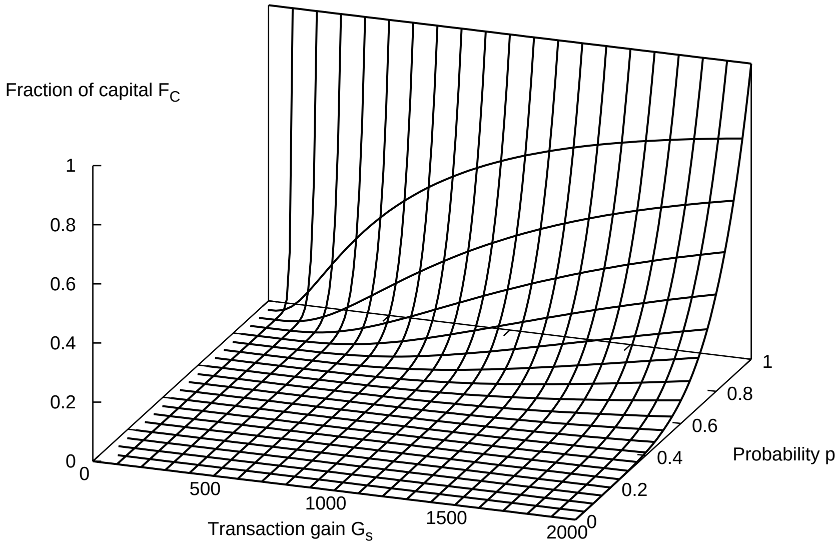


Fig. 7. Example of an agent's risk attitude expressed as a decision surface.

λ is interpreted as a factor of the relying party's risk attitude in the given transaction context, and in the graph of Fig.7 we have set $\lambda = 10000$. A low λ value is representative of a risk-taking behaviour because it increases the volume under the surface delimited by F_C (pushes the decision surface upwards in Figure 7), whereas a high λ value represents risk aversion because it reduces the volume under the surface (pushes the decision surface down).

Risk attitudes are relative to each individual, so the shape of the surface in Figure 7 only represents an example and will of course differ for each agent.

A particular transaction will be represented by a point in the 3D space of Figure 7 with coordinates $(G_{\rm s}, p, F_C)$. Because the surface represents an agent's risk attitude the agent will per definition accept a transaction for which the point is located underneath the decision surface, and will reject a transaction for which the point is located above the decision surface.

4 Reputation Models and Systems

Seen from the relying party's point of view, reputation scores can be computed based on own experience, on second hand referrals, or on a combination of both. In the jargon of economic theory, the term *private information* is used to describe first hand information resulting from own experience, and *public information* is used to describe publicly available second hand information, i.e. information that can be obtained from third parties.

Reputation systems are typically based on public information in order to reflect the community's opinion in general, which is in line with Def.4 of reputation. Private information that is submitted to a public reputation center is here considered as public information. An entity who relies on the reputation score of some remote party, is in fact trusting that party by implicitly trusting those who have rated that party, which in principle is *trust transitivity* as described in Sec.2.3.

This section describes reputation system architectures and various principles for computing reputation and trust measures. Some of the principles are used in commercial applications, whereas others have been proposed by the academic community.

4.1 Reputation Network Architectures

The technical principles for building reputation systems are described in this and the following section. The network architecture determines how ratings and reputation scores are communicated between participants in a reputation systems. The two main types are centralised and distributed architectures.

Centralised Reputation Systems In centralised reputation systems, information about the performance of a given participant is collected as ratings from other members in the community who have had direct experience with that participant. The central authority (reputation centre) that collects all the ratings typically derives a reputation score for every participant, and makes all scores publicly available. Participants can then use each other's scores, for example, when deciding whether or not to transact with a particular party. The idea is that transactions with reputable participants are likely to result in more favourable outcomes than transactions with disreputable participants.

Fig.8 below shows a typical centralised reputation framework, where A and B denote transaction partners with a history of transactions in the past, and who consider transacting with each other in the present.

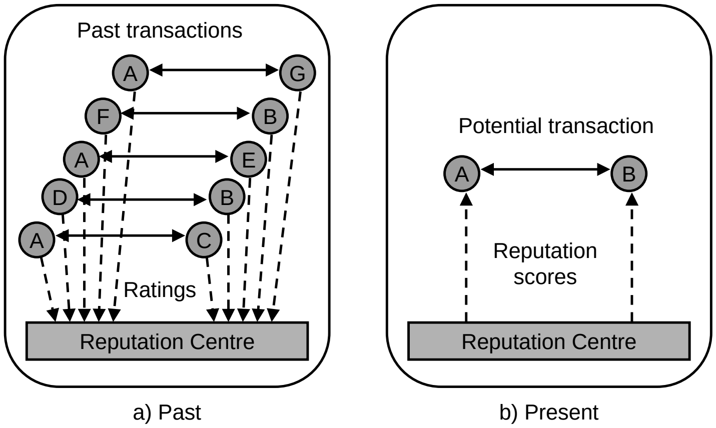


Fig. 8. General framework for a centralised reputation system

After each transaction, the agents provide ratings about each other's performance in the transaction. The reputation centre collects ratings from all the agents, and continuously updates each agent's reputation score as a function of the received ratings. Updated reputation scores are provided online for all the agents to see, and can be used by the agents to decide whether or not to transact with a particular agent. The two fundamental aspects of centralised reputation systems are:

a. *Centralised communication protocols* that allow participants to provide ratings about transaction partners to the central authority, as well as to obtain reputation scores of potential transaction partners from the central authority.
b. *A reputation computation engine* used by the central authority to derive reputation scores for each participant, based on received ratings, and possibly also on other information.

Distributed Reputation Systems There are environments where a distributed reputation system, i.e. without any centralised functions, is better suited than a centralised system. In a distributed system there is no central location for submitting ratings or obtaining reputation scores of others. Instead, there can be distributed stores where ratings can be submitted, or each participant simply records the opinion about each experience with other parties, and provides this information on request from relying parties. A relying party, who considers transacting with a given target party, must find the distributed stores, or try to obtain ratings from as many community members as possible who have had direct experience with that target party. This is illustrated in fig.9 below.

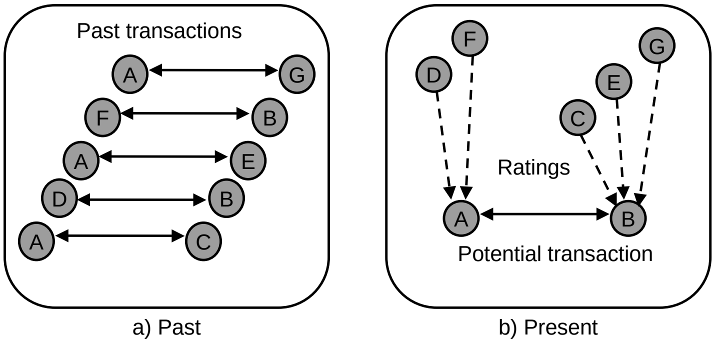


Fig. 9. General framework for a distributed reputation system

The relying party computes the reputation score based on the received ratings. In case the relying party has had direct experience with the target party, the experience from that encounter can be taken into account as private information, possibly carrying a higher weight than the received ratings. The two fundamental aspects of distributed reputation systems are:

a. *A distributed communication protocol* that allows participants to obtain ratings from other members in the community.

b. *A reputation computation method* used by each individual agent to derive reputation scores of target parties based on received ratings, and possibly on other information.

Peer-to-Peer (P2P) networks represent a environment well suited for distributed reputation management. In P2P networks, every node plays the role of both client and server, and is therefore sometimes called a *servent*. This allows the users to overcome their passive role typical of web navigation, and to engage in an active role by providing their own resources. There are two phases in the use of P2P networks. The first is the *search* phase, which consists of locating the servent where the requested resource resides. In some P2P networks, the search phase can rely on centralised functions. One such example is Napster[8] which has a resource directory server. In pure P2P networks like Gnutella[9] and Freenet[10], also the search phase is distributed. Intermediate architectures also exist, e.g. the FastTrack architecture which is used in P2P networks like KaZaA[11], grokster[12] and iMesh[13]. In FastTrack based P2P networks, there are nodes and supernodes, where the latter keep tracks of other nodes and supernodes that are logged onto the network, and thus act as directory servers during the search phase.

After the search phase, where the requested resource has been located, comes the *download phase*, which consists of transferring the resource from the exporting to the requesting servent.

P2P networks introduce a range of security threats, as they can be used to spread malicious software, such as viruses and Trojan horses, and easily bypass firewalls. There is also evidence that P2P networks suffer from free riding [4]. Reputation systems are well suited to fight these problems, e.g. by sharing information about rogue, unreliable or selfish participants. P2P networks are controversial because they have been used to distribute copyrighted material such as MP3 music files, and it has been claimed that content poisoning[14] has been used by the music industry to fight this problem. We do not defend using P2P networks for illegal file sharing, but it is obvious that reputation systems could be used by distributors of illegal copyrighted material to protect themselves from poisoning. Many authors have proposed reputation systems for P2P networks [2, 10–12, 18, 31, 34]. The purpose of a reputation system in P2P networks is to determine:

a. which servents are most reliable at offering the best quality resources, and
b. which servents provide the most reliable information with regard to (1).

In a distributed environment, each participant is responsible for collecting and combining ratings from other participants. Because of the distributed environment, it is often impossible or too costly to obtain ratings resulting from all interactions with a given agent. Instead the reputation score is based on a subset of ratings, usually from the relying party's "neighbourhood".

[8] http://www.napster.com/

[9] http://www.gnutella.com

[10] http://www.zeropaid.com/freenet

[11] http://www.kazaa.com

[12] http://www.grokster.com/

[13] http://imesh.com

[14] Poisoning music file sharing networks consists of distributing files with legitimate titles - and put inside them silence or random noise.

4.2 Simple Summation or Average of Reputation Ratings

The simplest form of computing reputation scores is simply to sum the number of positive ratings and negative ratings separately, and to keep a total score as the positive score minus the negative score. This is the principle used in eBay's reputation forum which is described in detail in [45]. The advantage is that anyone can understand the principle behind the reputation score, the disadvantage that it is primitive and therefore gives a poor picture participants' reputation score although this is also due to the way rating is provided.

A slightly more advanced scheme proposed in e.g. [49] is to compute the reputation score as the average of all ratings, and this principle is used in the reputation systems of numerous commercial web sites, such as Epinions, and Amazon.

Advanced models in this category compute a weighted average of all the ratings, where the rating weight can be determined by factors such as rater trustworthiness/reputation, age of the rating, distance between rating and current score etc.

4.3 Bayesian Reputation Systems

Bayesian systems have a solid mathematical foundation, and are based on computing reputation scores by statistical updating of binomial Beta or multinomial Dirichlet probability density functions (PDF). The *a posteriori* (i.e. the updated) reputation score is computed by combining the *a priori* (i.e. previous) reputation score with the new rating [26, 39–41, 53, 28]. Binomial reputation systems allow ratings to be expressed with two values, as either positive (e.g. *good*) or negative (e.g. *bad*). Multinomial reputation systems allow the possibility of providing ratings with graded levels such as e.g. *mediocre* - *bad* - *average* - *good* - *excellent*. In addition, multinomial models are able to distinguish between the case of polarised ratings (i.e. a combination of strictly good and bad ratings) and the case of only average ratings. The ability to indicate when ratings are polarised can provide valuable clues to the user in many situations. Multinomial reputation systems therefore provide great flexibility when collecting ratings and providing reputation scores.

Multinomial Bayesian reputation systems allow ratings to be provided over k different levels which can be considered as a set of k disjoint elements. Let this set be denoted as $\Lambda = \{L_1, \ldots L_k\}$, and assume that ratings are provided as votes on the elements of Λ. This leads to a Dirichlet probability density function over the k-component random probability variable $p(L_i),\ i = 1 \ldots k$ with sample space $[0, 1]^k$, subject to the simple additivity requirement $\sum_{i=1}^{k} p(L_i) = 1$.

The Dirichlet distribution with prior captures a sequence of observations of the k possible outcomes with k positive real rating parameters $r(L_i),\ i = 1 \ldots k$, each corresponding to one of the possible levels. In order to have a compact notation we define a vector $\vec{p} = \{p(L_i) \mid 1 \leq i \leq k\}$ to denote the k-component probability variable, and a vector $\vec{r} = \{r_i \mid 1 \leq i \leq k\}$ to denote the k-component rating variable.

In order to distinguish between the *a priori* default base rate, and the *a posteriori* ratings, the Dirichlet distribution must be expressed with prior information represented as a base rate vector $\vec{a}$ over the state space.

Let $\Lambda = \{L_1, \dots L_k\}$ be a state space consisting of k mutually disjoint elements which can be rating levels. Let $\vec{r}$ represent the rating vector over the elements of Λ and let $\vec{a}$ represent the base rate vector over the same elements. The reputation score is defined in terms of the expectation value of each random probability variable corresponding to the rating levels. This provides a sound mathematical basis for combining ratings and for expressing reputation scores. The probability expectation of any of the k random probability variables can be written as:

$$\mathrm{E}(p(L_i) \mid \vec{r}, \vec{a}) = \frac{r(L_i) + Ca(L_i)}{C + \sum_{i=1}^{k} r(L_i)} \ . \tag{2}$$

The *a priori* weight C will normally be set to $C = 2$ when a uniform distribution over binary state spaces is assumed. Selecting a larger value for C will result in new observations having less influence over the Dirichlet distribution. The combination of the base rate vector $\vec{a}$ and the *a priori* weight C can in fact represent specific *a priori* information provided by a domain expert or by another reputation system. It can be noted that it would be unnatural to require a uniform distribution over arbitrary large state spaces because it would make the sensitivity to new evidence arbitrarily small. The value of C determines the approximate number of votes needed for a particular level to influence the probability expectation value of that level from 0 to 0.5

A general reputation system allows for an agent to rate another agent or service, with any level from a set of predefined rating levels. Some form of control over what and when ratings can be given is normally required, such as e.g. after a transaction has taken place, but this issue will not be discussed here. Let there be k different discrete rating levels. This translates into having a state space of cardinality k for the Dirichlet distribution. Let the rating level be indexed by i. The aggregate ratings for a particular agent y are stored as a cumulative vector, expressed as:

$$\vec{R}_y = (R_y(L_i) \mid i = 1 \dots k) \ . \tag{3}$$

Each new discrete rating of agent y by an agent x takes the form of a trivial vector $\vec{r}_y^{\,x}$ where only one element has value 1, and all other vector elements have value 0. The index i of the vector element with value 1 refers to the specific rating level. The previously stored vector $\vec{R}$ is updating by adding the newly received rating vector $\vec{r}$.

Agents (and in particular human agents) may change their behaviour over time, so it is desirable to give relatively greater weight to more recent ratings. This can be achieved by introducing a longevity factor $\lambda \in [0, 1]$, which controls the rapidity with which old ratings are aged and discounted as a function of time. With $\lambda = 0$, ratings are completely forgotten after a single time period. With $\lambda = 1$, ratings are never forgotten.

Let new ratings be collected in discrete time periods. Let the sum of the ratings of a particular agent y in period t be denoted by the vector $\vec{r}_{y,t}$. More specifically, it is the sum of all ratings $\vec{r}_y^{x}$ of agent y by other agents x during that period, expressed by:

$$\vec{r}_{y,t} = \sum_{x \in M_{y,t}} \vec{r}_y^{x} \tag{4}$$

where $M_{y,t}$ is the set of all agents who rated agent y during period t.

Let the total accumulated ratings (with aging) of agent y after the time period t be denoted by $\vec{R}_{y,t}$. The new accumulated rating after time period $t+1$ is expressed as:

$$\vec{R}_{y,(t+1)} = \lambda \cdot \vec{R}_{y,t} + \vec{r}_{y,(t+1)}, \text{ where } 0 \leq \lambda \leq 1 \; . \tag{5}$$

Eq.(5) represents a recursive updating algorithm that can be executed every period for all agents. A reputation score applies to member agents in a community M. Before any evidence is known about a particular agent y, its reputation is defined by the base rate reputation which is the same for all agents. As evidence about a particular agent is gathered, its reputation will change accordingly.

The most natural representation of reputation scores is in the form of the probability expectation values of each element in the state space. The expectation value for each rating level can be computed with Eq.(2). Let $\vec{R}$ represent a target agent's aggregate ratings. The vector $\vec{S}$ defined by:

$$\vec{S}_y : \left(S_y(L_i) = \frac{R_y(L_i) + Ca(L_i)}{C + \sum_{j=1}^{k} R_y(L_j)} ; | \; i = 1 \dots k \right) \; . \tag{6}$$

is the corresponding multinomial probability reputation score. As already stated, $C = 2$ is the value of choice, but larger value for the weight C can be chosen if a reduce influence of new evidence over the base rate is required.

The reputation score $\vec{S}$ can be interpreted as a multinomial probability measure expressing how a particular agent is expected to behave in future transactions. It can easily be verified that

$$\sum_{i=1}^{k} S(L_i) = 1 \; . \tag{7}$$

The multinomial reputation score can for example be visualised as columns, which would clearly indicate if ratings are polarised. Assume for example 5 levels:

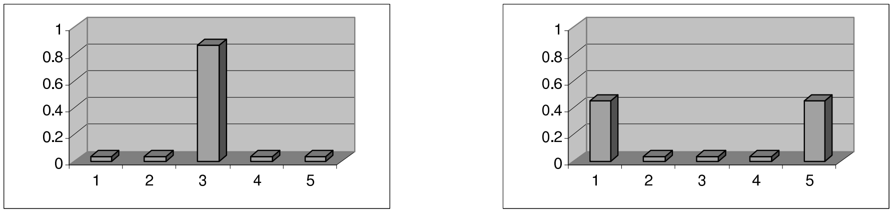
$$L_1 : \text{Mediocre}, \quad L_2 : \text{Bad}, \quad L_3 : \text{Average}, \quad L_4 : \text{Good}, \quad L_5 : \text{Excellent}. \tag{8}$$

We assume a default base rate distribution. Before any ratings have been received, the multinomial probability reputation score will be equal to $1/5$ for all levels. We consider two different cases where 10 ratings are received. In the first case, 10 *average* ratings are received, which translates into the concentric probability reputation score of Fig.10.a. In the second case, 5 mediocre and 5 excellent ratings are received, which translates into the polarized probability reputation score of Fig.10.b.

While informative, the multinomial probability representation can require considerable space to be displayed on a computer screen. A more compact form can be to express the reputation score as a single value in some predefined interval. This can be done by assigning a point value ν to each rating level i, and computing the normalised weighted point estimate score σ.

Assume e.g. k different rating levels with point values evenly distributed in the range [0,1], so that $\nu(L_i) = \frac{i-1}{k-1}$. The point estimate reputation is then computed as:

$$\sigma = \sum_{i=1}^{k} \nu(L_i) S(L_i) \; . \tag{9}$$



(a) After 10 average ratings

(b) After 5 mediocre and 5 excellent ratings (polarised case)

Fig. 10. Reputation scores resulting from average and from polarised ratings

However, this point estimate removes information, so that for example the difference between the average ratings and the polarised ratings of Fig.10.a and Fig.10.b is no longer visible. The point estimates of the reputation scores of Fig.10.a and Fig.10.b are both 0.5, although the ratings in fact are quite different. A point estimate in the range [0,1] can be mapped to any range, such as 1-5 stars, a percentage or a probability.

Bootstrapping a reputation system to a stable and conservative state is important. In the framework described above, the base rate distribution $\vec{a}$ will define initial default reputation for all agents. The base rate can for example be evenly distributed, or biased towards either a negative or a positive reputation. This must be defined by those who set up the reputation system in a specific market or community.

Agents will come and go during the lifetime of a market, and it is important to be able to assign new members a reasonable base rate reputation. In the simplest case, this can be the same as the initial default reputation that was given to all agents during bootstrap.

However, it is possible to track the average reputation score of the whole community, and this can be used to set the base rate for new agents, either directly or with a certain additional bias.

Not only new agents, but also existing agents with a standing track record can get the dynamic base rate. After all, a dynamic community base rate reflects the whole community, and should therefore be applied to all the members of that community.

The aggregate reputation vector for the whole community at time t is computed as:

$$\vec{R}_{M,t} = \sum_{y_j \in M} \vec{R}_{y,t} \tag{10}$$

This vector then needs to be normalised to a base rate vector as follows:

Definition 7 (Community Base Rate). *Let $\vec{R}_{M,t}$ be an aggregate reputation vector for a whole community, and let $S_{M,t}$ be the corresponding multinomial probability reputation vector which can be computed with Eq.(6). The community base rate as a function of existing reputations at time $t+1$ is then simply expressed as the community score at time t:*

$$\vec{a}_{M,(t+1)} = \vec{S}_{M,t}. \tag{11}$$

The base rate vector of Eq.(11) can be given to every new agent that joins the community. In addition, the community base rate vector can be used for every agent every time their reputation score is computed. In this way, the base rate will dynamically reflect the quality of the market at any one time.

If desirable, the base rate for new agents can be biased in either negative or positive direction in order to make it harder or easier to enter the market.

As an example we consider the following sequence of varying ratings:

Periods 1 - 10:	L1 Mediocre
Periods 11 - 20:	L2 Bad
Periods 21 - 30:	L3 Average
Periods 31 - 40:	L4 Good
Periods 41 - 50:	L5 Excellent

The longevity factor is $\lambda = 0.9$ as before, and the base rate is dynamic. The evolution of the scores of each level as well as the point estimate are illustrated in Fig.11.

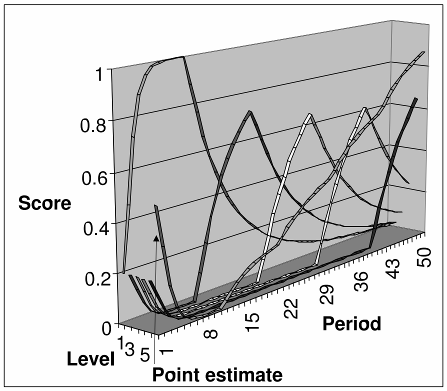


Fig. 11. Scores and point estimate during sequence of varying ratings

In Fig.11 the multinomial reputation scores change abruptly between each sequence of 10 periods. The point estimate first drops as the score for L1 increase during the first 10 periods. After that the point estimate increases relatively smoothly during the subsequent 40 periods. Because of the dynamic base rate, the point estimate will eventually converge to 1.

5 Applications and Examples

5.1 The PGP Trust Model

The software encryption tool PGP (Pretty Good Privacy) [55] provides support for managing public keys and public-key certificates. The trustworthiness of imported keys and their owners is derived using PGP's particular trust model.

Trust is is applied to three different aspects which are *"Owner Trust"* which corresponds to trust in the owner of a public key, *"Signature Trust"* which corresponds to trust in received certificates, and *"Key Validity"* which corresponds to trust in a public key, where each trust type can take discrete trust values, as indicated below.

$$\begin{array}{l}\text{Owner Trust}\\ \text{Signature Trust}\end{array}\left\{\begin{array}{l}\textit{always trusted}\\ \textit{usually trusted}\\ \textit{not trusted}\\ \textit{unknown trust}\end{array}\right. \qquad \text{Key Validity}\left\{\begin{array}{l}\textit{complete}\\ \textit{marginal}\\ \textit{undefined}\end{array}\right.$$

A user's private key(s) is/are stored in a table called the 'Secret Key Ring'. Keys stored here are used for signing messages and to decrypt received encrypted messages. A table called the 'Public Key Ring' is used to store other users' public keys together with the trust parameters Key Validity and Owner trust for each key. Keys stored here are used for encrypting messages sent to other users and to verify signed messages received from them.

When a new public key is received and introduced through a certificate PGP first checks that the Key Validity of the key used for signing the certificate is *complete*, otherwise the certificate is ignored. After having accepted a certificate its Signature Trust gets the Owner Trust value of the user who signed it. When a key has one or more certificates, the accumulated Signature Trust values determine the Key Validity of the key according to the skepticism level. By default PGP requires one *always trusted* or two *usually trusted* signatures in order to assign *complete* Key Validity to the received public key, but these parameters can be tuned by the user according to his or her trust attitude. An insufficient number of *always trusted* or *usually trusted* signatures results in *marginal* Key Validity, and a key received without even a *usually trusted* signature gets *undefined* Key Validity.

Only the Key Validity is automatically computed by PGP, not the Owner Trust. PGP therefore asks the user how much he or she trusts the owner for introducing new keys, and this decision is purely subjective. The Key Validity and the Owner Trust parameters represent confidential information that is not communicated to other users.

After having defined the Key Validity and Owner Trust for a particular key, PGP allows the user to sign and add it to the Public Key Ring. The user can now introduce it to others who will evaluate this key in exactly the same way as describe above. The various elements (with their corresponding discrete trust parameters in brackets) are illustrated in Fig.12.

Because trust parameters are subjective it is not meaningful to share the Public Key Ring with others. Furthermore it is only meaningful to express trust in someone you know, theoretically limiting the number of keys that anyone can store on the Public Key Ring to the number of people he or she actually knows. Current usage however shows that this design assumption is wrong; many people fill their Public Key Rings with keys of people whom they have never met and with whom they have never communicated. Unfortunately this practice destroys PGP's trust management and reduces PGP to a purely mechanical encryption tool.

PGP users can in principle follow whatever certification practice they want but they are of course expected to be convinced that the key is authentic, or in PGP terms that

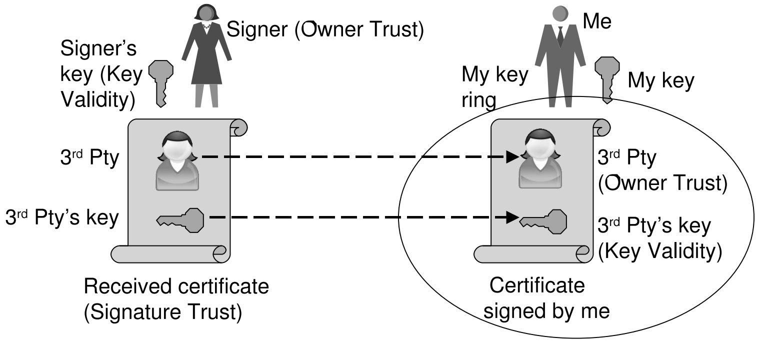


Fig. 12. The PGP trust model

the Key Validity is considered to be *complete*, before issuing a certificate and this can be considered as an informal certificate policy. PGP was primarily built as an e-mail encryption tool for creating a secure channel between people who know each other or who can establish an indirect trust path between each other, and for that purpose it has been extremely successful.

5.2 Web Page Ranking

The early web search engines such as Altavista simply presented every web page that matched the key words entered by the user, which often resulted in too many and irrelevant pages being listed in the search results. Altavista's proposal for handling this problem was to offer advanced ways to combine keywords based on binary logic. This was too complex for users, and therefore did not provide a good solution.

PageRank proposed by Page *et al.* (1998) [43] represents a way of ranking the best search results based on a page's score according to a specific metric. Roughly speaking, PageRank computes the score for any Web page as the sum of the normalised weights of hyperlinks pointing to it, where a normalised hyperlink weight is determined by the score of the page containing the hyperlink, divided by the total number of hyperlinks from that page. This can be described as a trust system, because the total set of hyperlinks form transitive trust chains that can be used as a basis for deriving a relative trust measures for each page. A single hyperlink to a given web page can be seen as a unidirectional trust relationship between the source and the target page. Google's search engine[15] is based on the PageRank algorithm, and the rapidly rising popularity of Google at the cost of Altavista was obviously caused by the superior search results that the PageRank algorithm delivered. The definition of the PageRank algorithm from Page *et al.* (1998) [43] is given below:

Definition 8 (PageRank). Let P be a set of hyperlinked web pages and let u and v denote web pages in P. Let $N^{-}(u)$ denote the set of web pages pointing to u and

[15] http://www.google.com/

let $N^{+}(v)$ denote the set of web pages that v points to. Let s be some vector over P corresponding to a distribution of initial score such that $\sum_{u \in P} s(u) = 1$. Then, the rank of a web page u is:

$$r(u) = d\, s(u) \quad + \quad (1-d) \sum_{v \in N^{-}(u)} \frac{r(v)}{|N^{+}(v)|}\ , \tag{12}$$

In [43] it is recommended that d be chosen such that $d = 0.15$. The first term in Eq.(12) gives rank value based on initial score. The second term gives rank value as a function of normalised weights of hyperlinks pointing at u. The algorithm of Def.8 must be iterated over the whole Web until the scores for all Web pages stabilise.

The PageRank algorithm provides an algorithmic representation of the *"random surfer model"*, i.e. the value $r(u)$ represents the probability of arriving at Web page u by randomly surfing the Web. Intuitively, because of the very large total number of hyperlinked Web pages in the Internet, this probability value is very close to zero for any random web page.

According to Def.8, $r(u) \in [0, 1]$, but the PageRank values that Google provides to the public are scaled to the range [0,10]. We will denote the public PageRank of a page u as $PR(u)$. This public PageRank measure can be viewed for any web page using Google's toolbar which is a plug-in to the MS Internet Explorer. Although Google do not specify exactly how the public PageRank is computed, it is widely conjectured that it measured on a logarithmic scale with base close to 10. An approximate expression for computing the public PageRank could for example be:

$$PR(u) = l + \log_{10} r(u) \tag{13}$$

where l is a constant that defines the cut-off value, so that only pages with $r(u) > 10^{-l}$ will be listed by Google. A typical value is $l = 11$.

It is not publicly known how the source rank vector s is defined, but it would be natural to distribute it over the root web pages of all domains weighted by the cost of buying each domain name. Assuming that the only way to improve a page's PageRank is to buy domain names, Clausen (2004) [9] shows that there is a lower bound to the cost of obtaining an arbitrarily good $PR(u)$ for a Web page u.

Without specifying many details, Google state that the PageRank algorithm they are using also takes other elements into account, with the purpose of making it difficult or expensive to deliberately influence PageRank.

In order to provide a semantic interpretation of a PageRank value, a hyperlink can be seen as a positive rating of the page it points to. Negative ratings do not exist in PageRank so that it is impossible to blacklist web pages with the PageRank algorithm of Eq.(12) alone. Before Google with it's PageRank algorithm entered the search engine arena, some webmasters would promote web sites by filling web pages with large amounts of commonly used search key words as invisible text or metadata in order for the page to have a high probability of being listed by a search engine no matter what the user searched for. The PageRank algorithm compensates for problem because a high *R* is also needed in addition to matching key words in order for a page to be presented to the user.

The growing importance of having a high score in search engines has made many owners of Web sites very restrictive with placing hyperlinks to other websites, because outgoing hyperlinks normally result in decreased scores for Web pages on the own Web site. The very existence of search engines thus had the inevitable effect of interfering with the structure of the Web.

The increasing popularity and economic importance of search engines has also lead to more damaging methods for artificially boosting the score of Web pages. One such example is the phenomenon called *link spam* which consists of placing many hyperlinks to the same Web page on open Web fora such as online discussion boards, guest books, weblogs and wikis. The motivation behind this attack is that search engines will give an increased score for the Web page that these hyperlinks point to.

In order to counter the link spam attacks Google announced in early 2005 that hyperlinks marked with the attribute `rel="nofollow"` would not influence the hyperlink target's score in the search engine's index. This is implemented as follows:

```
<a href="http://some-spammer-website.com"
rel="nofollow" >Click here!</a>
```

Most open Web fora now mark user-submitted hyperlinks this way by default, with no option to disable it by the users, and most search engines take it into account when computing scores. This is an example where a simple technical solution was able to solve a growing problem. However, it has negative side effects.

The increasing usage of `rel="nofollow"` in Web pages will have the effect that scores computed by Google and search engines no longer reflect the real structure of the Web, and removes the model more and more from the random surfer model. The random surfer follows any link, whereas search engines only follow those that are not marked by `rel="nofollow"`. A likely development is that most outgoing hyperlinks will be marked in this way in a selfish manner in order not to suffer decreased scores. The search engines will then face the problem of scarcity of cross links between Web sites, making the computed scores increasingly unreliable.

As a substitute for the hyperlinks, search engines need to use other types of evidence. An obvious source of information is the links that users actually select after a specific search. algorithms can for example be designed that increase the rank of a specific Web page when many people select the link to that page after a Web search. However, the value of this information is limited, because it only becomes available during searches, and does not reflect which Web pages people go to when not using search engines.

It would be more valuable for search engines to know the link to every page that people visit. By encouraging people to use toolbars, search engines can get precisely that information. A toolbar provides some value-added functionality to users, such as displaying the PageRank of every page the user visits. In return for this functionality, the engine is informed about every single Web page that the user visits. This architecture is illustrated in Fig.13 below.

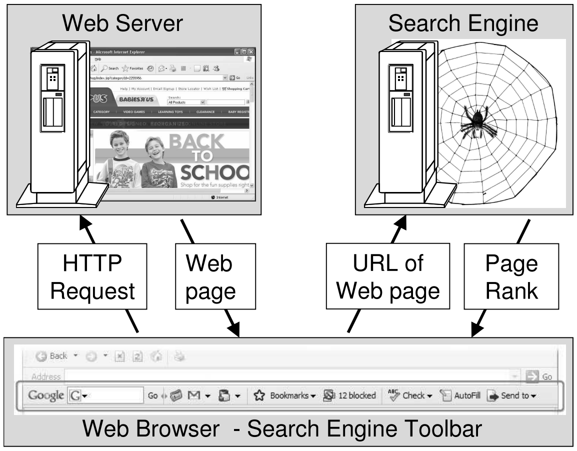


Fig. 13. Network architecture for search engine toolbars

Users often ignored that the toolbar provides this information. Constantly providing the search engine with information about Web pages visited by the user can be considered quite intrusive, and this functionality is usually also found in so-called spyware.

On the basis of information provided by toolbars, search engines are able to compute the probability that an intentional surfer will go to any particular Web page. This can be called the *intentional surfer model*, which represents an improvement over the random surfer model of the original PageRank algorithm.

However, it is likely that the current model is already under attack with the purpose of artificially increasing the ranking of certain Web pages. An obvious attack method is e.g. to install search engine toolbars on a large number of computers, and let programs automatically browse specific Websites. Google and other toolbar providers are aware of this potential problem, and usually registers each individual toolbar installation in order to identify possible "click spamming". It is still unclear to what degree click spamming already is or will be a problem in the near future. The "pay per click" business model is being abused through click spamming, and it is therefore to be expected that the intentional surfer model that bases rank on the number of clicks to Web pages already is under attack as well.

In general it can be observed that any new method for improving rank computation becomes the subject of new attacks as soon as it is implemented. The robustness and reliability of searching and Web navigation has become a cat-and-mouse game, similarly to that of traditional information security.

As a simple example of how a reputation system can be implemented in a general level we describe a simple reputation toolbar which can be installed on any browser. This allows the reputation score of any Web page to be visualised to the user, as well as the user to rate Web sites and Web pages. The toolbar communicates with a centralised server which keeps the reputation vectors of all Web pages. A Web page can be rated by the user with a discrete set of different levels, as described above. This architecture is illustrated in Fig.14

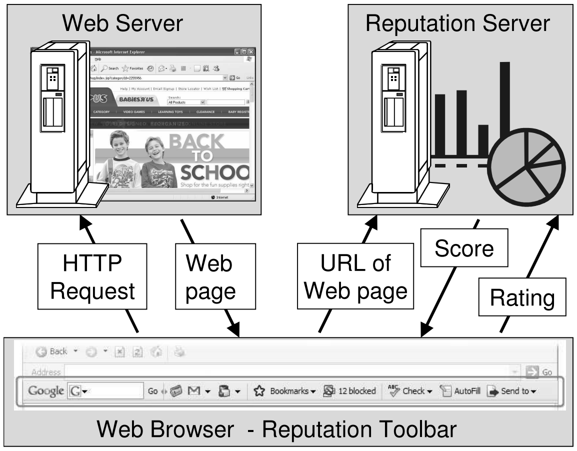


Fig. 14. Network architecture for reputation toolbars

While the browser is fetching a Web page, the reputation toolbar will query the reputation server about the reputation score of that Web page or Web site. The user is also invited to rate the same Web site through the toolbar. This rating is sent to the reputation server, and taken into account when computing the reputation score in the future.

The functionality of the reputation toolbar of Fig.14 can very well be integrated with a traditional search engine toolbar. The reputation scores can be taken into account for computing rank when presenting Web search results, or can be presented as a separate score for each search query result. In the latter case, the reputation server and the search engine do not need to be co-located. The reputation score can simply be fetched as part of a search query, either by the search engine itself, or by a shell on the client machine. The addition of a reputation system to the traditional search engine will allow the implementation of the *critical surfer model*, which represents an improvement over the current *random surfer* and the *intentional surfer* models, as illustrated in Fig.15.

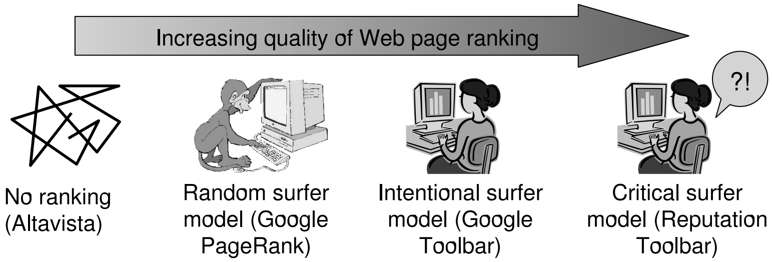

Increasing quality of Web page ranking
?!
No ranking (Altavista)
Random surfer model (Google PageRank)
Intentional surfer model (Google Toolbar)
Critical surfer model (Reputation Toolbar)

Fig. 15. Past, present and future Web ranking models

While the introduction of the PageRank algorithm represented a revolution in the quality of Web searches, there is still an untapped potential for improvement by integrating reputation systems with search engines.

5.3 The Slashdot Model and Hierarchic Reputation Systems

An approach that seems to work relatively well is that of meta-moderation used on Slashdot[16] which is a *"news for nerds"* message board started in 1997. More precisely it is a forum for posting articles and comments to articles. In the early days when the community was small, the signal to noise ratio was very high. As is the case with all mailing lists and discussion fora where the number of members grow rapidly, spam and low quality postings emerged to become a major problem, and this forced Slashdot to introduce moderation. To start with there was a team of 25 moderators which after a while grew to 400 moderators to keep pace with the growing number of users and the amount of spam that followed. In order to create a more democratic and healthy moderation scheme, automated moderator selection was introduced.

The moderation scheme actually consists of two moderation layers where M1 is for moderating comments to articles, and M2 is for moderating M1 moderators. The purpose of M1 is to be able to filter the good comments from the bad. The purpose of M2 is to address the issue of unfair moderations, or more precisely to sanction M1 moderators. Above M2 in the hierarchy are the staff of Slashdot with omnipotent powers to sanction any M1 or M2 moderator who is detected in abusing the system. Details of the Slashdot reputation systems are described in [27].

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

In a hierarchical reputation system, ratings occur at different levels, and scores can be computed for elements on each level. Here we describe a general approach to designing hierarchical reputation systems.

Service objects can have a reputation score based on ratings from service users. Users who provide ratings have a credit score based ratings from moderators. Moderators have a credit score based on ratings from Controllers who represent the top of the hierarchy. Users rate service objects positively or negatively based on direct experience with those services. A reputation score can be computed for each object as a function of those ratings. Moderators can rate users depending on whether they provide fair or unfair service ratings. A credit score can be computed for each user based on the user's fairness in rating services. The idea is that service ratings provided by discredited users will carry relatively less weight than service ratings provided by credited users, when the reputation scores for service objects are derived. Controllers, who for example can be representatives from the reputation centre, can rate moderators, and depending on the design, can also rate users. This model is illustrated in Fig.16.

[16] http://slashdot.org/

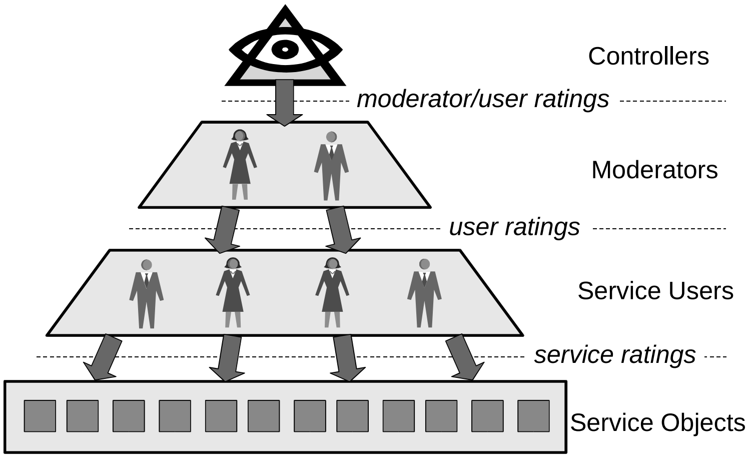


Fig. 16. Hierarchic model for reputation systems

The idea being this design is to spread the workload of providing ratings over all the service users, and provide a mechanisms for stabilising the system and sanctioning unfair raters. Design issues are for example the determination of the optimal Moderator/User and Controller/Moderator ratios, and defining adequate incentives for participants to contribute to the collaborative effort. From a purely rational viewpoint, a participant has little incentive to rate a service after the fact, because providing ratings benefits others, not oneself. A study from eBay [45] shows that 60.7% of the buyers and 51.7% of the sellers on eBay provided ratings about each other. Possible explanations for these relatively high values can for example be that providing reciprocal ratings simply is an expression of politeness. However lack of incentives for providing ratings is a general problem that needs special attention when designing reputation systems.

6 Discussion and Conclusion

The robustness of trust and reputation systems for resisting attacks and strategic manipulation is the critical factor for the success of this technology, and which currently is not being sufficiently addressed. Traditional security mechanisms can be used to achieve goals such as anonymity and integrity of ratings [19]. Identity and credentials management can be used to control when and by whom ratings can be provided, e.g. to prevent ballot stuffing [20]. The robustness of soft security mechanisms will thus depend on hard security mechanisms.

Social acceptance of trust and reputation systems is another critical factor, which many commercial systems have addressed and solved quite well. However, for the more widespread and general usage of these systems, social acceptance by all parties is an issue that needs to be considered.

Given that reputation systems used in commercial and online applications have serious vulnerabilities, it is obvious that the reliability of these systems sometimes is questionable. Assuming that reputation systems give unreliable scores, why then are they used? A possible answer to this question is that in many situations the reputation systems do not need to be robust because their value lies elsewhere. Resnick & Zeckhauser (2002) [45] consider two explanations in relation to eBays reputation system: (a) Even though a reputation system is not robust it might serve its purpose of providing an incentive for good behaviour if the participants think it works, and (b) even though the system might not work well in the statistical normative sense, it may function successfully if it swiftly reacts against bad behaviour (called *"stoning"*) and if it imposes costs for a participant to get established (called *"label initiation dues"*).

Given that some online reputation systems are far from being robust, it is obvious that the organisations that run them have a business model that is relatively insensitive to their robustness. It might be that the reputation system serves as a kind of social network to attract more people to a web site, and if that is the case, then having simple rules for participating is more important than having strict rules for controlling participants' behaviour. Any reputation system with user participation will depend on how people respond to it, and must therefore be designed with that in mind. Another explanation is that, from a business perspective, having a reputation system that is not robust can be desirable if it generally gives a positive bias. After all, commercial web stores are in the business of selling, and positively biased ratings are more likely to promote sales than negative ratings.

Whenever the robustness of a reputation system is crucial, the organisation that runs it should take measures to protect the stability of the system and robustness against attacks. This can for example be by including routine manual control as part of the scheme, such as in Epinions' case when selecting Category Lead reviewers, or in Slashdot's case where Slashdot staff are omnipotent moderators. Exceptional manual control will probably always be needed, should the system come under heavy attack. Another important element is to keep the exact details of the computation algorithm and how the system is implemented confidential (called *"security by obscurity"*), such as in the case of Epinions, Slashdot and Google. Ratings are usually based on subjective judgement, which opens up the Pandora's box of unfair ratings, but if ratings can be based on objective criteria it would be much simpler to achieve high robustness.

The rich literature growing around trust and reputation systems for Internet transactions, as well as the implementation of reputation systems in successful commercial application, give a strong indication that this is an important technology. The early commercial and live implementations were, and still are, based on relatively simple schemes, whereas a multitude of different systems with advanced features are continuously being proposed by the academic community. Some of the advanced schemes are slowly finding their way into real implementations as more experience is gained with this type of technology.

Designing and implementing robust trust and reputation systems represents a formidable challenge, and the long term acceptance of the Internet as a reliable platform for supporting open markets and communities depends on the success of this endeavour. To have effective and pervasive trust management on the Internet is like finding the holy grail because the value of the Internet would increase manifold. How to make it happen is therefore an extremely important research problem for the global Internet community.

References

1. A. Abdul-Rahman and S. Hailes. Supporting Trust in Virtual Communities. In *Proceedings of the Hawaii International Conference on System Sciences*, Maui, Hawaii, 4-7 January 2000 2000.
2. K. Aberer and Z. Despotovic. Managing trust in a peer-2-peer information system. In Henrique Paques, Ling Liu, and David Grossman, editors, *Proceedings of the Tenth International Conference on Information and Knowledge Management (CIKM01)*, pages 10–317. ACM Press, 2001.
3. M.D. Abrams. Trusted System Concepts. *Computers and Security*, 14(1):45–56, 1995.
4. E. Adar and B.A. Huberman. Free Riding on Gnutella. *First Monday (Peer-reviewed Journal on the Internet)*, 5(10):8, Otober 2000.
5. Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proceedings of the 1996 IEEE Conference on Security and Privacy*, Oakland, CA, 1996.
6. V. Cahill, B. Shand, E. Gray, et al. Using Trust for Secure Collaboration in Uncertain Environments. *Pervasive Computing*, 2(3):52–61, July-September 2003.
7. M. Carbone, M. Nielsen, and V. Sassone. A Formal Model for Trust in Dynamic Networks. In *Proc. of International Conference on Software Engineering and Formal Methods (SEFM'03)*, Brisbane, September 2003.
8. B. Christianson and W. S. Harbison. Why Isn't Trust Transitive? In *Proceedings of the Security Protocols International Workshop*. University of Cambridge, 1996.
9. 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.
10. F. Cornelli et al. Choosing Reputable Servents in a P2P Network. In *Proceedings of the eleventh international conference on World Wide Web (WWW'02)*. ACM, May 2002.
11. E. Damiani et al. A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks. In *Proceedings of the 9th ACM conference on Computer and Communications Security (CCS'02)*, pages 207–216. ACM, 2002.
12. D. Fahrenholtz and W. Lamesdorf. Transactional Security for a Distributed Reputation Management System. In *Proceedings of the Third International Conference on E-Commerce and Web Technologies (EC-Web)*, volume LNCS 2455, pages 214–223. Springer, September 2002.
13. R. Falcone and C. Castelfranchi. How trust enhances and spread trust. In *Proceedings of the 4th Int. Workshop on Deception Fraud and Trust in Agent Societies, in the 5th International Conference on Autonomous Agents (AGENTS'01)*, May 2001.
14. R. Falcone and C. Castelfranchi. Social Trust: A Cognitive Approach. In C. Castelfranchi and Y.H. Tan, editors, *Trust and Deception in Virtual Societies*, pages 55–99. Kluwer, 2001.
15. L.C. Freeman. Centrality on Social Networks. *Social Networks*, 1:215–239, 1979.
16. D. Gambetta. Can We Trust Trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–238. Basil Blackwell. Oxford, 1990.
17. T. Grandison and M. Sloman. A Survey of Trust in Internet Applications. *IEEE Communications Surveys and Tutorials*, 3, 2000.
18. M. Gupta, P. Judge, and M. Ammar. A reputation system for peer-to-peer networks. In *Proceedings of the 13th international workshop on Network and operating systems support for digital audio and video (NOSSDAV)*, 2003.
19. R. Ismail, C. Boyd, A. Jøsang, and S. Russel. Strong Privacy in Reputation Systems. In *Proceedings of the 4th International Workshop on Information Security Applications (WISA)*, Jeju Island, Korea, August 2003.

20. R. Ismail, C. Boyd, A. Jøsang, and S. Russel. An Efficient Off-Line Reputation Scheme Using Articulated Certificates. In *Proceedings of the Second International Workshop on Security in Information Systems (WOSIS-2004)*, 2004.
21. ISO. *ISO/IEC IS17799 - Information technology – Code of practice for information security management*. ISO/IEC, 2005.
22. A. Jøsang. The right type of trust for distributed systems. In C. Meadows, editor, *Proc. of the 1996 New Security Paradigms Workshop*. ACM, 1996.
23. A. Jøsang. Trust-Based Decision Making for Electronic Transactions. In L. Yngström and T. Svensson, editors, *Proceedings of the 4th Nordic Workshop on Secure Computer Systems (NORDSEC'99)*. Stockholm University, Sweden, 1999.
24. A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, 9(3):279–311, June 2001.
25. A. Jøsang. Probabilistic Logic Under Uncertainty. In *The Proceedings of Computing: The Australian Theory Symposium (CATS2007), CRPIT Volume 65*, Ballarat, Australia, January 2007.
26. A. Jøsang and R. Ismail. The Beta Reputation System. In *Proceedings of the 15th Bled Electronic Commerce Conference*, June 2002.
27. 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.
28. A. Jøsang and Haller J. Dirichlet Reputation Systems. In *The Proceedings of the International Conference on Availability, Reliability and Security (ARES 2007)*, Vienna, Austria, April 2007.
29. 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.
30. A. Jøsang and S. Pope. Semantic Constraints for Trust Tansitivity. In S. Hartmann and M. Stumptner, editors, *Proceedings of the Asia-Pacific Conference of Conceptual Modelling (APCCM) (Volume 43 of Conferences in Research and Practice in Information Technology)*, Newcastle, Australia, February 2005.
31. 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*, Budapest, May 2003.
32. K. Krukow and M. Nielsen. From Simulations to Theorems: A Position Paper on Research in the Field of Computational Trust. In *Proceedings of the Workshop of Formal Aspects of Security and Trust (FAST 2006)*, Ontario, Canada, August 2006.
33. R. Levien. *Attack Resistant Trust Metrics*. PhD thesis, University of California at Berkeley, 2004.
34. C.Y. Liau et al. Efficient Distributed Reputation Scheme for Peer-to-Peer Systems. In *Proceedings of the 2nd International Human.Society@Internet Conference (HSI)*, volume LNCS 2713, pages 54–63. Springer, 2003.
35. D.W. Manchala. Trust Metrics, Models and Protocols for Electronic Commerce Transactions. In *Proceedings of the 18th International Conference on Distributed Computing Systems*, 1998.
36. P.V. Marsden and N. Lin, editors. *Social Structure and Network Analysis*. Beverly Hills: Sage Publications, 1982.
37. D.H. McKnight and N.L. Chervany. The Meanings of Trust. Technical Report MISRC Working Paper Series 96-04, University of Minnesota, Management Information Systems Reseach Center, 1996.
38. Merriam-Webster. *Merriam-Webster Online*. Available from http://www.m-w.com/, accessed June 2007.

39. L. Mui, M. Mohtashemi, and C. Ang. A Probabilistic Rating Framework for Pervasive Computing Environments. In *Proceedings of the MIT Student Oxygen Workshop (SOW'2001)*, 2001.

40. L. Mui, M. Mohtashemi, C. Ang, P. Szolovits, and A. Halberstadt. Ratings in Distributed Systems: A Bayesian Approach. In *Proceedings of the Workshop on Information Technologies and Systems (WITS)*, 2001.

41. L. Mui, M. Mohtashemi, and A. Halberstadt. A Computational Model of Trust and Reputation. In *Proceedings of the 35th Hawaii International Conference on System Science (HICSS)*, 2002.

42. OASIS. *Conformance Requirements for the OASIS Security Assertion Markup Language (SAML) V2.0,* Committee Draft. Organization for the Advancement of Structured Information Standards, 15 January 2005.

43. L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford Digital Library Technologies Project, 1998.

44. L. Rasmusson and S. Janssen. Simulated Social Control for Secure Internet Commerce. In Catherine Meadows, editor, *Proceedings of the 1996 New Security Paradigms Workshop*. ACM, 1996.

45. P. Resnick and R. Zeckhauser. Trust Among Strangers in Internet Transactions: Empirical Analysis of eBay's Reputation System. In M.R. Baye, editor, *The Economics of the Internet and E-Commerce*, volume 11 of *Advances in Applied Microeconomics.* Elsevier Science, 2002.

46. J. Sabater and C. Sierra. REGRET: A reputation model for gregarious societies. In *Proceedings of the 4th Int. Workshop on Deception, Fraud and Trust in Agent Societies, in the 5th Int. Conference on Autonomous Agents (AGENTS'01)*, pages 61–69, Montreal, Canada, 2001.

47. J. Sabater and C. Sierra. Reputation and Social Network Analysis in Multi-Agent Systems. In *Proceedings of the First Int. Joint Conference on Autonomous Agents & Multiagent Systems (AAMAS)*, July 2002.

48. J. Sabater and C. Sierra. Social ReGreT, a reputation model based on social relations. *SIGecom Exchanges*, 3.1:44–56, 2002.

49. J. Schneider et al. Disseminating Trust Information in Wearable Communities. In *Proceedings of the 2nd International Symposium on Handheld and Ubiquitous Computing (HUC2K)*, September 2000.

50. G.J. Simmons. An introduction to the mathematics of trust in security protocols. In *Proceedings of the 1993 Computer Security Foundations Workshop*, pages 121–127. IEEE Computer Society Press, Los Alamitos, CA, USA, 1993.

51. S. Tadelis. Firm Reputation with Hidden Information. *Economic Theory*, 21(2):635–651, 2003.

52. O.E. Williamson. Calculativeness, Trust and Economic Organization. *Journal of Law and Economics*, 36:453–486, April 1993.

53. A. Withby, A. Jøsang, and J. Indulska. Filtering Out Unfair Ratings in Bayesian Reputation Systems. *The Icfain Journal of Management Research*, 4(2):48–64, 2005.

54. C.-N. Ziegler and G. Lausen. Spreading Activation Models for Trust Propagation. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service (EEE '04)*, Taipei, March 2004.

55. P.R. Zimmermann. *The Official PGP User's Guide.* MIT Press, 1995.

About the author

Audun Jøsang is an Associate Professor at Queensland University of Technology in Brisbane, Australia. His research focuses on trust and reputation systems in addition to information security. Audun received his PhD from the Norwegian University of Science and Technology in 1998, has a MSc in Information Security from Royal Holloway College, University of London, and a BSc in Telematics from the Norwegian Institute of Technology.