A Trust Model for an Open, Decentralized Reputation System

Andreas Gutscher

Universität Stuttgart,
Institute of Communication Networks and Computer Engineering,
D-70569 Stuttgart, Germany
gutscher@ikr.uni-stuttgart.de

Summary. The use of reputation systems has been proposed for various applications, e. g., to estimate the trustworthiness of sellers and buyers in electronic transactions. Reputation systems collect opinions of users about properties of certain services, subjects and other users and evaluate these opinions. It is important that the results of reputation systems are consistent with intuitive expectations of its users, which highly depends on the properties of the underlying trust model. The trust model defines the representation of the trust values as well as the computation of trust values for derived trust relations.

We propose a new sophisticated computational model of trust which seamlessly integrates authentication verification into the trust evaluation process and which is suitable especially for open, decentralized reputation systems. It consists of definitions of trust and authentication relations, inference rules and three downward compatible trust calculi. It is therefore possible to represent and evaluate trust values with different levels of detail. The model reflects all relevant aspects and properties of trust and authentication relations and therefore avoids any counterintuitive effects.[1]

1 Introduction

1.1 Reputation Systems

A reputation system is an approach to systematically evaluate opinions of online community members on various issues (e. g., products, services, events, etc.) and their opinions on the trustworthiness of other community members.

Reputation systems first collect and combine all relevant opinions, draw conclusions about the trustworthiness of all opinions from the subjective perspective of a given user and calculate the trustworthiness of all opinions referring to certain issues. Then, all opinions referring to a particular issue are combined according to their trustworthiness, and the result is returned to the

[1] This work was funded by the German Research Foundation (DFG) through the Center of Excellence (SFB) 627.

requesting user or application, where it can be used to make a decision, e. g., to recommend the highest ranked restaurant.

The use of reputation systems has been proposed for various applications, for example to validate the trustworthiness of sellers and buyers in online auctions, to detect free-riders in peer-to-peer networks and to ensure the authenticity of signature keys in a *web of trust* (e. g., PGP [1]).

Evaluating large sets of different and possibly contradictory opinions is a non-trivial yet crucial process. The *trust model* of a reputation system represents the core concepts of the system. It defines all assumptions on the properties of trust relations and describes how to calculate the resulting trust values.

1.2 Related Work

There exists a large number of propositions for computational models and systems which intend to support humans, agents and applications in deciding whether or not to interact with other parties based on the accumulated opinions of others. However, the field of proposed solutions is quite diversified, so that even surveys [2, 3] have difficulties to cover the whole range from collaborative filtering systems [4], recommender and reputation systems, risk and trust management system [5], deterministic and probabilistic trust models, formal and logic frameworks [6] for trust, distrust [7], uncertainty and forgiveness [8] to experimental sociological studies [9]. Therefore, only selected propositions can be covered here.

Stephen Paul Marsh [10, 7] was one of the first researchers to formalize the concept and various aspects of trust and to represent them by a mathematical model which can be evaluated and used for the implementation of artificial trusting agents.

A trust model that emerged from probability theory is the Dempster-Shafer model [11]. It assigns probabilities to sets and subsets of events. Two values, *belief* and *plausibility*, define the upper and the lower bound of the probability corresponding to a given set of interest. With them, it is possible to express a degree of uncertainty. The *Dempster's rule of combination* defines how to combine the opinions of two independent observers. This rule has been criticized by many researchers for its property to create counterintuitive results, and several alternative combination rules have been proposed [12].

Thomas Beth et al. [13] proposed a model for estimating the trustworthiness of entities in open networks on the basis of recommendations of mediators. An initial trust value is calculated from the number of positive and negative past experiences and direct trust is distinguished from recommendation trust.

Audan Jøsang [14, 15] has developed a mathematical model called "subjective logic." The opinion space corresponds to the area of an *opinion triangle*, the angles represent full *belief*, *disbelief* and *ignorance* (which is equivalent to the representation of trust values in the Dempster-Shafer model). Jøsang defines a set of operators to calculate with opinions, e. g., operators for the

conjunction and disjunction of two opinions as well as consensus and recommendation operators. However, this model and all other trust models with non-distributive operators are not applicable to arbitrary trust structures but only to *directed series-parallel graphs* [16].

One of the currently most widely deployed trust models for public key validation is the model used in the PGP *Web of Trust* [1]. Trust and authenticity statements can be expressed and distributed via digitally signed certificates. The strength of trust and key authenticity can be expressed by discrete trust levels. A set of rules defines how to derive new trust and authenticity relations starting from an initial set of trusted relations specified by the user. A limit for the length of the trust chains can only be specified globally by the validator, but not by the issuer of the trust certificates. It has been shown in [17] that this model can produce counterintuitive results.

Ueli Maurer [18] has proposed a model for trust and authenticity relations for public key authentication in PKIs and introduces recommendation levels for trust relations. Unlike the models using operators to combine two opinions, Maurer proposes to calculate the resulting trust value on the basis of probability calculus instead and avoids thus the above-mentioned trust graph evaluation problem. However, the trust model is limited to public key authentication, and it has been criticised to make the restricting implicit assumption, that each principal holds exactly one key pair [19].

An important yet difficult task is the evaluation and validation of trust models. Several design principles and validation criteria for trust models have been proposed in [20, 19] and [17], but there is no consensus on whether all trust models should follow these principles or whether trust models for different applications may have different requirements [21].

1.3 Contributions

Due to the above mentioned problems and limitations of existing trust models we propose a new trust model (basing on Maurer's trust model [18]), which tries to overcome these issues and which is better suited especially for open decentralized reputation systems. The model integrates public key authenticity verification into the trust model, it avoids any counterintuitive effects, it may be used to evaluate arbitrary trust structures, it supports multiple keys (and identity descriptions) per user, it enables the signer of a trust certificate to limit the length trust chains, it does not force users to stick to a limited number of discrete trust values and clearly defines the semantic of the trust values. Moreover, it offers three different trust calculi basing on the same relations and inference rules but offering different levels of detail. The trust model can therefore serve as a sophisticated replacement for currently used trust models in various open decentralized reputation and public key validation systems (e. g., the PGP trust model).

The remainder of this paper is organized as follows. Section 2 describes the scenario and attacker model, in section 3 we discuss properties of trust

relations. An overview on the trust model is given in section 4. In section 5 the trust and authenticity relations and in section 6 the inference rules of the model are described. In section 7 three trust calculi are proposed. We discuss our approach in section 8 and conclude in section 9.

2 Problem Description

We consider an open system without a central authority. *Entities* (the users of the reputation system, e. g., humans, agents, etc.) can join and leave the system at any time and may use different identities (or pseudonyms). Entities can generate multiple asymmetric *key pairs*, sign statements and verify signatures. We assume, that entities have some kind of distinct names, addresses, attributes, etc. so that it is possible to compose *descriptions* which refer unambiguously to the current identity of an entity. Several different descriptions may refer to the same identity.

Entities can formulate *ratings*. A rating is a statement describing the subjective opinion of an entity on some *issues* (e. g., "I believe that pizzas from X are very tasty"). Each issue corresponds to one or more *capabilities* which are considered necessary to formulate a useful rating. An entity cannot definitely determine whether or to which extend an other entity possess a particular capability, but it can determine the initial *trustworthiness* of the entity with respect to this capability. The trustworthiness is a measure for the *subjective estimation* of whether the other entity has this capability (competence and goodwill), based on own experience and knowledge. Similarly, an entity can make subjective estimations about the authenticity of public keys. Entities may use different, application-dependent strategies to determine these estimations (e. g., [13]), however, a discussion is out of scope. Entities can sign ratings as well as trust and authenticity statements and publish those certificates. All entities can retrieve and analyze all published certificates.

Each entity (or a trusted reputation service) can *evaluate* own trust and authenticity statements together with all public trust and authenticity certificates from other entities in order to determine the trustworthiness of all entities for all capabilities and the authenticity of all public keys. Then, the trustworthiness of all ratings can be determined, and finally all ratings for the same issue can be merged according to their respective trustworthiness. This merged rating can then serve as basis for decision-making.

Note that we consider the uncertainty which originates from the subjective estimations as predominant factor for the reliability and usefulness of the result. Therefore, the trust model is designed to capture and trace the impact of uncertainty which originates from the subjective estimations and to determine the most likely conclusion. We do *not* try to measure, whether the system is resistant to attacks against the dissemination of trust or authentica-

tion information (e. g., by forcing entities to revoke already issued certificates) [2].

Attackers may try to influence the result of the evaluations by publishing arbitrary rating, trust and authenticity certificates as regular entities, but we assume that attackers cannot prevent other entities from publishing their certificates. Cryptographic mechanisms are assumed to be secure, and private keys are never disclosed.

3 Trust

In general, trust is often described as *the subjective belief of someone in the character, ability, strength, reliability, honesty or truth of someone or something* [3]. In this paper, however, we adopt the following, more technical working definitions (based on [22]):

Trust (or a trust relation) *is a unidirectional relation between a truster and a trustee expressing the strong belief of the truster that the trustee will behave as expected with respect to a particular capability within a particular context.*

Trustworthiness (or a trust value) *is a quantitative measure of the strength of a trust relation representing the subjective estimation of the likelihood that the trustee will behave as expected with respect to a particular capability within a particular context.*

We do not discuss finer grained classifications for trust (e. g., distinguish *competence* and *goodwill*) as they do not have direct implications on our model.

Trust relations have a number of properties, which must be properly reflected by trust models in order to avoid counterintuitive results.

Specificity Trust is specific for a particular *capability* c within a particular context[3]. Trust for a particular capability does in general not imply trust for other capabilities. This can be illustrated by the following example: The fact, that Alice trusts Bobs for giving useful recommendations on recent movies does *not* imply that she trusts him for giving medical advice. In our model, the capability c may either be

- the pre-defined capability c_{PKI} representing the capability, that the trustee will honestly and carefully verify that a given description of an entity refers to the holder of a particular public key, or
- an arbitrary application specific capability, e. g., c_1, c_2, etc.

[2] In the latter case you might wish to have a look at the trust model proposed in [19]

[3] in the following, we simplifyingly use the term "capability" only

Direct and Indirect Trust Trust relations can be divided into *direct* and *indirect trust* relations. *Direct trust* (or *functional trust*) represents the opinion of the truster, that the trustee *has* the specified capability, e. g., "Bob trusts Carol to be a good dentist." *Indirect trust* (or *recommender trust*) represents the opinion of the truster, that the trustee will give useful *recommendations* for this capability, e. g., "Alice trusts Bob to recommend good dentists." Note that Alice does not express her opinion on Bobs qualities as dentist, Bob does not even have to be a dentist at all in order to give useful recommendations. For indirect trust relations, we can further distinguish recommendations with different numbers of *recommendation hops* (or *levels*) $h > 0$:

- An indirect trust relation with $h = 1$ expresses, that the truster trusts the trustee for recommending a third entity which has the specified capability.
- An indirect trust relation with $h = 2$ expresses, that the truster trusts the trustee for recommending a third entity which is trustworthy for recommending a forth entity which has the specified capability.
- etc.

A value of $h = 0$ denotes a direct trust relation. Note that values for h are normally very small (typically $h \leq 2$).

Symmetry Trust relations are in general *not symmetric*. The fact, that Alice trusts Bob does not imply that Bob trusts Alice. Trust relations must thus be modeled as *unidirectional* relations.

Reflexivity Trust relations are in general *not reflexive*, i. e., an entity does not always trust itself. This apparently implausible property can be illustrated by the following example: Alice might consider herself to be not trustworthy with respect to the capability of doing surgery (because she knows that she has no medical skills).

Transitivity Many trustmodels are based on the assumption that trust relations are transitive. Although it seems to be intuitive and correct to rely on recommendations of trustworthy entities in some cases, we emphasize that trust relations are *not necessarily always transitive*. This can be illustrated by the following example: Alice beliefs that Bob is gullible but honest, and she trusts Bob for lending money. Bob considers Carol to be trustworthy for lending money and recommends Carol to Alice. However, Alice beliefs, that Bob is not able to judge whether Carol is honest or not. Thus, in this case it is reasonable for Alice not to trust Carol for lending money. This apparent contradiction disappears if we distinguish more clearly between direct and indirect trust [23]. Trust shows transitive properties *only for specific combinations of direct and indirect trust relations*. These conditions and the parameters of the resulting trust relations are definded by the transitive trust inference rule in section 6.1.

Time Variability Trust may change over time, either due to new experiences or due to inactivity of the trustee. Therefore, the usual certificate update and recovery mechanisms (time-stamps, validity periods, certificate revocation lists, etc.) should be deployed. These mechanisms (as well as their problems) have been well-investigated and will be omitted in the following for simplicity.

4 Trust Model Overview

Our trust model is composed of four building blocks and it allows to choose between three calculi (see Figure 1). The basic two blocks are independent of the chosen calculus. They define all existing *trust and authentication relations* (section 5) and describe *inference rules* to combine the relations (section 6).

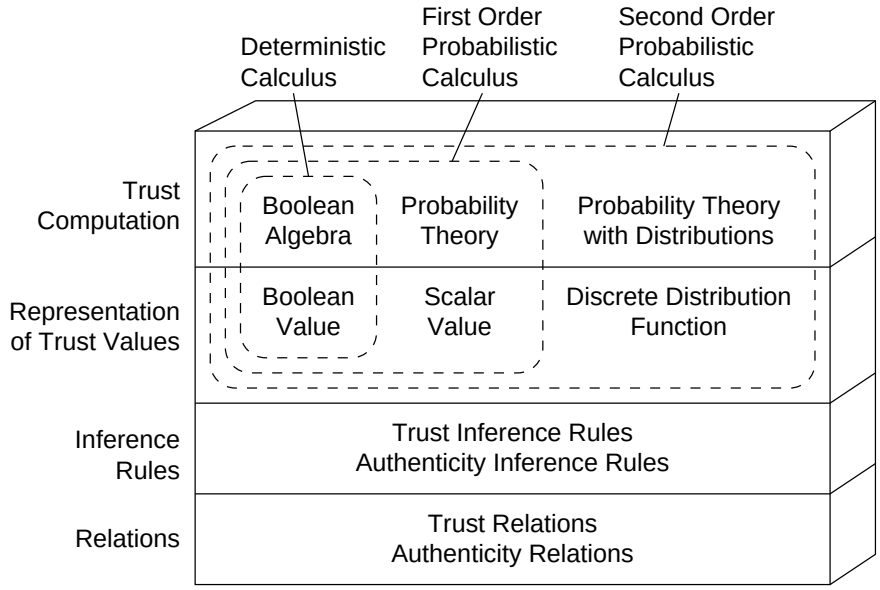

Fig. 1. Trust model overview

The other two blocks are calculus-specific. They describe how to *represent trust values* and how to *compute the trust values* of derived trust relations (section 7). For simple applications, which do not need to distinguish multiple trust levels, the simple *deterministic calculus* will be sufficient. The *first-order probabilistic trust calculus* operates on trust values which correspond to probabilities. The most flexible calculus is the *second-order probabilistic trust calculus* Here, trust values can be expressed by discrete probability distribution functions. Note that the format of *ratings* is not defined within the trust model because it may be application specific.

5 Trust and Authenticity Relations

Users decide whether they trust other entities, e. g., a human being, an organization, a server etc. However, in order to share these trust opinions with other users they typically have to be exchanged via digitally signed trust certificates. In these certificates, users have to reference the entities by unique identifiers or descriptions. In open systems either the public keys of the entities may serve as unique identifiers or some kind of description may be used, e. g., first and last names or host names, postal or network addresses, profession and affiliation or even photos. Humans often prefer to use descriptions because they consider it much easier to associate an entity with a description than with it's public key.

Thus, the authenticity of a public key or a description of an entity can constitute a prerequisite for the evaluation of trust certificates, because it may be necessary to validate that the key used to sign a trust certificate belongs to a trusted entity or that public keys and descriptions in these certificates belong to the same entity. At the same time, the trustworthiness of an entity can constitute a prerequisite for the evaluation of the authenticity of a public key or a description of an entity because it may be necessary to validate the trustworthiness of the entity that claims that a public key or a description belongs to a particular entity. Therefore, it does not make sense to first evaluate all trust relations and then to decide whether the authenticity of the public keys can be validated. Neither does it make sense to first evaluate the authenticity of all public keys and to consider the trust relations afterwards. Instead, trust and authenticity relations have to be evaluated in an integrated process. Therefore, public keys, descriptions of entities and various relations between them are an integral part of our trust model in order to seamlessly integrate the authenticity verification with the trust computation.

In the following, we define nine trust and authenticity relations. Relations issued by a public key represent signed certificates and can therefore be exchanged with other users, whereas relations issued by an entity serve for local evaluation only.

5.1 Trust Relations

Trust Relation Between Two Entities An entity E_A can express trust in another entity E_B for the capability c for h recommendation hops with the trust relation:

$$E_A : Trust(E_B, c, h) \qquad (h \geq 0) \tag{1}$$

Trust Referring to a Public Key An entity E_A can express trust in the *entity possessing the private key* corresponding to the public key K_B for the capability c and h recommendation hops with the trust relation:

$$E_A : Trust(K_B, c, h) \qquad (h \geq 0) \tag{2}$$

Trust Certificate Referring to a Public Key A *trust certificate referring to a public key* expresses (similarly to equation 2) that the owner of the private key corresponding to the public key K_A trusts the entity possessing the private key corresponding to the public key K_B for the capability c and h recommendation hops:

$$K_A : Trust(K_B, c, h) \qquad (h \geq 0) \tag{3}$$

Trust Referring to a Description An entity E_A can express trust in the entity matching the description D_B for the capability c and h recommendation hops with the trust relation:

$$E_A : Trust(D_B, c, h) \qquad (h \geq 0) \tag{4}$$

Trust Certificate Referring to a Description A *trust certificate referring to an entity description* expresses (similarly to equation 4) that the owner of the private key corresponding to the public key K_A trusts the entity matching the description D_B for the capability c and h recommendation hops.

$$K_A : Trust(D_B, c, h) \qquad (h \geq 0) \tag{5}$$

5.2 Authenticity Relations

Authenticity of Public Keys An entity E_A can express its belief that the entity E_B is the owner of the private key corresponding to the public key K_B with the authenticity relation:

$$E_A : Auth(K_B, E_B) \tag{6}$$

Authenticity of Entity Descriptions An entity E_A can express its belief that the description D_B refers non-ambiguously to the entity E_B with the authenticity relation:

$$E_A : Auth(D_B, E_B) \tag{7}$$

Relationship between Public Keys and Descriptions An entity E_A can express its belief that the description D_B refers non-ambiguously to the entity which is the owner of the private key corresponding to the public key K_B with the authenticity relation:

$$E_A : Auth(K_B, D_B) \tag{8}$$

Identity Certificates An *identity certificate* expresses (similarly to equation 8) that the owner of the private key corresponding to the public key K_A beliefs that the description D_B refers non-ambiguously to the entity which is the owner of the private key corresponding to the public key K_B:

$$K_A : Auth(K_B, D_B) \tag{9}$$

6 Trust and Authenticity Inference Rules

The following set of rules describes the logic of the trust model. These rules define whether and which relations one can derive from a set of given relations, i. e., which conclusions result from a set of given relations.

It is important to distinguish clearly between relations from different origins: *First-hand relations* are relations which have been issued by users based only on own experience and knowledge and which are independent from other issued relations. *Second-hand relations* are relations which have been derived from other relations using inference rules. Note that only first-hand relations may be published in certificates. Second-hand relations must not be disseminated to other users.

The evaluation process starts with an *initial set* V of first-hand relations, which consists of all first-hand relations expressed by the user himself and all available published certificates[4]. The inference rules can then be applied repeatedly to the initial set expanded by all previously derived second-hand relations. This procedure is repeated until no more new relations can be derived. The set of relations consisting of the initial set V and all relations which can be derived from V is denoted by $\bar{V}$.

6.1 Trust Inference

Trust Inference for Lower Hops Indirect trust for more than one hop implies indirect trust for fewer hops:

$$A : Trust(B, c, h) \quad \wedge \quad h > 1 \quad \Rightarrow \quad A : Trust(B, c, h-1)$$

The truster A can be an entity (E_A) or a public key (K_A). The trustee B can be an entity (E_B), a public key (K_B) or a description (D_B).

Transitive Trust Inference The following rule describes the *transitivity* property of trust relations. It defines in which cases two trust relations can be combined in order to derive a new trust relation from the truster of the first relation to the trustee of the second relation. This rule summarizes two cases. It describes how direct trust can be derived from an indirect and a direct trust relation ($h_2 = 0$), and how indirect trust can be derived from two indirect trust relations ($h_2 > 0$):

$$\begin{aligned} & A : Trust(B, c, h_1) \quad \wedge \quad B : Trust(C, c, h_2) \\ \wedge \quad & ((h_2 = 0 \quad \wedge \quad h_1 > 0) \quad \vee \quad (h_2 > 0 \quad \wedge \quad h_1 > 1)) \\ & \Rightarrow \quad A : Trust(C, c, min(h_1 - 1, h_2)) \end{aligned}$$

The truster A can be an entity (E_A) or a public key (K_A). The second relation can be a trust relation or a trust certificate, i. e., B can be an entity (E_B) or a public key (K_B). The final trustee C can be an entity (E_C), a public key (K_C) or a description (D_C).

[4] relations with trust value *no trust* can be removed from V immediately

Trust in Entities, Keys and Descriptions If an entity is trusted then an authentic key of the entity can be trusted, too (and vice versa):

$$E_A : Trust(E_C, c, h) \quad \wedge \quad E_A : Auth(K_C, E_C) \quad \Rightarrow \quad E_A : Trust(K_C, c, h)$$
$$E_A : Trust(K_C, c, h) \quad \wedge \quad E_A : Auth(K_C, E_C) \quad \Rightarrow \quad E_A : Trust(E_C, c, h)$$

If an entity is trusted then an authentic description of the entity can be trusted, too (and vice versa):

$$E_A : Trust(E_C, c, h) \quad \wedge \quad E_A : Auth(D_C, E_C) \quad \Rightarrow \quad E_A : Trust(D_C, c, h)$$
$$E_A : Trust(D_C, c, h) \quad \wedge \quad E_A : Auth(D_C, E_C) \quad \Rightarrow \quad E_A : Trust(E_C, c, h)$$

If a key of an entity is trusted then an authentic description of the entity can be trusted, too (and vice versa):

$$E_A : Trust(K_C, c, h) \quad \wedge \quad E_A : Auth(K_C, D_C) \quad \Rightarrow \quad E_A : Trust(D_C, c, h)$$
$$E_A : Trust(D_C, c, h) \quad \wedge \quad E_A : Auth(K_C, D_C) \quad \Rightarrow \quad E_A : Trust(K_C, c, h)$$

6.2 Authenticity Inference

Local Authenticity Inference If two corresponding authenticity relations are known, a third authenticity relation can be derived:

$$E_A : Auth(K_C, D_C) \quad \wedge \quad E_A : Auth(K_C, E_C) \quad \Rightarrow \quad E_A : Auth(D_C, E_C)$$
$$E_A : Auth(K_C, D_C) \quad \wedge \quad E_A : Auth(D_C, E_C) \quad \Rightarrow \quad E_A : Auth(K_C, E_C)$$
$$E_A : Auth(K_C, E_C) \quad \wedge \quad E_A : Auth(D_C, E_C) \quad \Rightarrow \quad E_A : Auth(K_C, D_C)$$

Authenticity Inference with Identity Certificates If an entity directly trusts a certification authority for issuing identity certificates (c_{PKI}), then the entity can consider the authenticity statements published in identity certificates signed by this certification authority to be valid:

$$E_A : Trust(K_B, c_{\mathrm{PKI}}, 0) \quad \wedge \quad K_B : Auth(K_C, D_C) \quad \Rightarrow \quad E_A : Auth(K_C, D_C)$$

7 Trust Calculi

Users associate each trust relation r with a trust value $t = \mathrm{conf}(r)$. Propositions for valid trust values reach from *positive trust* ("I trust X") via *no trust* ("I have no indication that X is trustworthy", also called *ignorance* or *uncertainty*) to *negative trust* ("I distrust X"). We started from the assumption of an open system, i. e., users may discard their current identity whenever they earn bad reputation and rejoin later with a new, clean identity. Therefore, we propose to refrain from using *negative trust* and to use instead *no trust* as the lowest trust value, which will be used as default value for strangers.

7.1 Deterministic Calculus

This calculi is based on boolean algebra. It is very simple to implement and intended for applications which do not need to distinguish multiple trust levels. Trust values are represented by boolean values: $t = 0$ represents *no trust* and $t = 1$ represents *full trust*.

The trust values of derived trust relations can be determined as follows: A derived relation r is *fully trusted* ($t = 1$) if and only if it can be derived from an initial set V of *fully trusted* trust relations (i. e., if $r \in \bar{V}$), else r is *not trusted* ($t = 0$). Note that it is sufficient to find a single trust path (i. e., a minimal set of trusted relations and a sequence of inference steps to derive r) in order to decide that r is *fully trusted*. Other (even trusted) opinions suggesting *no trust* do not reduce the trust value of r.

7.2 First-Order Probabilistic Trust Calculus

The *first-order probabilistic trust calculus* is based on probability theory and has similarities to the probabilistic model in [18]. The deterministic calculus is a special case of the first-order probabilistic trust calculus.

Trust values of relations are represented by real numbers within the interval $[0, 1]$. The lowest possible value $t = 0$ represents *no trust* and the highest possible value $t = 1$ represents *full trust*. Trust values are interpreted as probabilities, which represent the subjective estimation of the probability that the concerning relation is valid.

As we interprete trust values as probabilities, the computation of trust values of derived relations is performed according to probability theory. We consider the following random experiment: Each relation r_i ($i = 1, 2, \ldots, n$) of the initial set V is considered valid with a probability equal to its trust value $t_i = \mathrm{conf}(r_i)$. The resulting trust value t of a derived relation r is then equal to the probability that r can be derived from the set of valid relations

$$t = P\{r \in \bar{V}\}$$

An algorithm for the calculation of $P\{r \in \bar{V}\}$ can be constructed on the basis of the following consideration: Each relation r_i can either be valid (with probability $\mathrm{conf}(r_i)$) or invalid (with probability $1 - \mathrm{conf}(r_i)$). Therefore, we can construct 2^n different subsets of valid relations of V ("*possible worlds*"), which we denote by S_j ($j = 1, 2, \ldots, 2^n$). The probability, that the world S_j is the existing world, is

$$w_j = \prod_{r_i \in S_j} \mathrm{conf}(r_i) \cdot \prod_{r_i \notin S_j} 1 - \mathrm{conf}(r_i)$$

The trust value of r is the sum of the probabilities w_j of all worlds S_j, in which r can be derived from S_j:

$$t = \sum_{r \in \bar{S}_j} w_j$$

An algorithm for an efficient implementation of this computation has been proposed by Ueli Maurer [18].

7.3 Second-Order Probabilistic Trust Calculus

The *second-order probabilistic trust calculus* makes use of discrete probability distributions to represent trust values. The first-order probabilistic trust calculus is a special case of the second-order probabilistic trust calculus.

A trust value can be represented by a discrete probability distribution function, which allows to express uncertainty. The discrete probability distribution can be represented by a finite list of trust values t^i with an associated probability value p^i.

$$t = \{(t^1, p^1), (t^2, p^2), \ldots, (t^k, p^k)\} \qquad t^i,\ p^i \in [0,1], \quad \sum_{i=1}^{k} p^i = 1$$

The lowest possible value $t = \{(0,1)\}$ represents *no trust* and the highest possible value $t = \{(1,1)\}$ represents *full trust*.

The trust value of a derived relation can be calculated as follows: We consider all possible combinations of all trust values of the first relation r_1 $t_1^1, t_1^2, \ldots t_1^{k_1}$ with all trust values of the second relation r_2 $t_2^1, t_2^2, \ldots t_2^{k_2}$ etc. with all trust values of the last relation r_n $t_n^1, t_n^2, \ldots t_n^{k_n}$ ($\prod_{i=1}^{n} k_i$ combinations). For each combination $(t_1^1, t_2^1, \ldots t_n^1), (t_1^2, t_2^1, \ldots, t_n^1), \ldots, (t_1^{k_1}, t_2^{k_2}, \ldots, t_n^{k_n})$ we perform the same computation as in the case of the first-order probabilistic trust calculus. Finally, we construct the discrete probability distribution function: For each of the previous combination we get a resulting trust value from the computation. The associated probability value is computed as product of the probability values associated with the involved trust values of the relations from the initial set. If the trust value computation for two or more combinations return the same trust value, then the trust-probability-pairs can be merged by adding the associated probabilities.

The expectation $E[t] = \sum_i t^i p^i$ of a distribution function can be used if a scalar trust value is required, e. g., to compare two distribution functions or to merge ratings.

8 Discussion

Trust models for reputation systems should not be designed to *emulate* the sometimes irrational behaviour of humans. Instead, they should *improve* the ability of users to evaluate opinions and to come to the most beneficial decision. Therefore, it is not useful to check, whether agents using a particular

trust model show the same behaviour as humans (i. e., whether they would pass a "trust touring test" [21]). Instead, we believe that it is important to validate, that the models fulfill functional requirements, that they comply with rational principles and that the results do not show counterintuitive effects.

Therefore, we validate our model on the basis of some relevant principles (e. g., proposed in [20, 19] and [17]) and on aspects, which have been criticized in other trust models.

8.1 Features

The model is able to *evaluate arbitrary trust structures*. It supports *multiple key pairs* and *multiple descriptions* per entity, and is able to express *uncertainty* in trust opinions (with the second-order probabilistic trust calculus). It is based on a *sound mathematical basis* (probability theory), and the *meaning* of trust values is well-defined (trust value corresponds to a probability) and can directly be used in risk analysis. The model allows to specify the number of *recommendation hops* for each indirect trust relation. It *integrates authentication* of public keys and it supports *three downward compatible calculi* with different representations of trust.

8.2 Intuitive Behaviour

The model does not violate any of the following *rational intuitive principles*: *Adding* arbitrary trust or authentication relations does not decrease trust. *Concatenation* of trust relations does not increase trust. Trust relations, which are *not part of any valid trust path* have no influence on the resulting trust value. Trust based on *multiple recommendations* from a single source is not higher than that from independent sources.

8.3 Complexity and Implementation Aspects

The complexity of evaluation algorithms and other implementation aspects are of course important factors. However, we believe that the first (and apparently not yet satisfactorily completed) step is to find trust models which offer the required functionality and which show no counterintuitive behaviour. The question, whether computation complexity is a prior issue or not, may depend on the application. Even if a trust model turns out to be unsuitable for efficient implementation, there can be room for optimizations and simplifications and it may be as well a valuable reference to validate simpler estimation algorithms.

The deterministic calculus shows low complexity. The first-order probabilistic trust calculus can lead to a high complexity if the number of relations in the valid trust paths is high. The complexity can be reduced significantly by summarizing parallel relations and concatenations before the final trust value computation. The second-order probabilistic trust calculus will have a

high complexity if the number of trust values per distribution function is high. However, we believe that users will seldom require more than two trust values per distribution function to represent their opinions.

Incremental evaluation (i. e., reusing parts of the previous evaluations when new trust or authentication relations become available) is possible and efficient for the search of valid trust paths, but not for the computation of the resulting trust value.

First *prototypical implementations* in Java and in stored procedures of a relational database have shown, that the performance highly depends on the chosen data structures and that optimizations (e. g., as proposed in [18]) have the potential to speed up the computation by orders of magnitude.

9 Conclusion and Outlook

We have presented a new sophisticated computational model of trust for the evaluation of trust and authentication relations from the view of a user. Due to the integrated authenticity verification of public keys used to sign trust certificates, it is especially suitable for open, decentralized reputation systems and other applications, in which the authenticity of public keys is not verified otherwise. We discussed properties of trust relations and proposed a new trust model. It defines all possible trust and authenticity relations and their parameters, inference rules to draw conclusions and three downward compatible trust calculi which allow for representations of trust values with a different level of detail and complexity. Finally, we have shown that it provides a multitude of important functional aspects, that it complies with requirements for intuitive trust evaluation results and discussed complexity and implementation issues.

Some remaining challenges are algorithms and optimizations for the efficient computation of trust values as well as the discussion and evaluation of further principles of trust models.

References

1. Ashley, J.M., Copeland, M., Grahn, J., Wheeler, D.A.: The GNU Privacy Handbook. The Free Software Foundation. (1999)
2. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. In: Decision Support Systems. (2007)
3. Grandison, T., Sloman, M.: A survey of trust in internet application. IEEE Communications Surveys & Tutorials **3**(4) (2000)
4. Good, N., Schafer, J.B., Konstan, J.A., Borchers, A., Sarwar, B., Herlocker, J., Riedl, J.: Combining collaborative filtering with personal agents for better recommendations. In: Proceedings of the Sixteenth National Conference on Artificial Intelligence. (1999) 439–446
5. Suryanarayana, G., Taylor, R.N.: A survey of trust management and resource discovery technologies in peer-to-peer applications. Technical Report UCI-ISR-04-6, Institute for Software Research, University of California (2004)

6. Demolombe, R.: Reasoning about trust: A formal logical framework. In: Proceedings of the Second International Conference of Trust Management (iTrust 2004). (2004) 291–303
7. Marsh, S., Dibben, M.R.: Trust, Untrust, Distrust and Mistrust – An Exploration of the Dark(er) Side. In Herrmann, P., Issarny, V., Shiu, S., eds.: Proceedings of Third iTrust International Conference (iTrust 2005), Paris, France, May 23-26, 2005. Volume 3477., Springer (May 2005) 17–33
8. Vasalou, A., Pitt, J.: Reinventing forgiveness: A formal investigation of moral facilitation. In: Proceedings of the Third International Conference of Trust Management (iTrust 2005). (2005) 146–160
9. Jonker, C.M., Schalken, J.J.P., Theeuwes, J., Treur, J.: Human experiments in trust dynamics. In: Proceedings of the Second International Conference of Trust Management (iTrust 2004). (2004) 206–220
10. Marsh, S.P.: Formalising Trust as a Computational Concept. PhD thesis, Department of Mathematics and Computer Science, University of Stirling (1994)
11. Shafer, G.: A Mathematical Theory of Evidence. Princeton Univ. Press (1976)
12. Sentz, K., Ferson, S.: Combination of Evidence in Dempster-Shafer Theory (2002)
13. Beth, T., Borcherding, M., Klein, B.: Valuation of Trust in Open Networks. In: Proceedings 3rd European Symposium on Research in Computer Security (ESORICS) 1994, Springer-Verlag (1994) 3–18
14. Jøsang, A.: Artificial Reasoning with Subjective Logic (1997)
15. Jøsang, A., Knapskog, S.: A Metric for Trusted Systems. In: Proceedings 21st National Security Conference 1998. (1998)
16. Jøsang, A., Gray, E., Kinateder, M.: Simplification and analysis of transitive trust networks. In: Web Intelligence and Agent Systems Journal. (2006) 139–161
17. Kohlas, R., Maurer, U.: Confidence Valuation in a Public-key Infrastructure Based on Uncertain Evidence. In: In the proceedings of Public Key Cryptography 2000. Volume 1751 of Lecture Notes in Computer Science. (January 2000) 93–112
18. Maurer, U.: Modelling a Public-Key Infrastructure. In Bertino, E., ed.: Proc. 1996 European Symposium on Research in Computer Security (ESORICS' 96). Volume 1146 of Lecture Notes in Computer Science., Springer-Verlag (1996) 325–350
19. Reiter, M.K., Stubblebine, S.: Toward acceptable metrics of authentication. In: Proceedings of IEEE Symposium on Security and Privacy. (1997) 10–20
20. Sun, Y.L., Yu, W., Han, Z., Liu, K.J.R.: Information Theoretic Framework of Trust Modeling and Evaluation for Ad Hoc Networks. In: IEEE Journal on Selected Areas in Communications, Volume 24, Issue 2. (Feb 2006) 305–317
21. Langheinrich, M.: When Trust Does Not Compute The Role of Trust in Ubiquitous Computing. Workshop on Privacy at Ubicomp 2003 (October 2003)
22. Gambetta, D. In: Can We Trust Trust? Basil Blackwell (1988) 213–237 Reprinted in electronic edition from Department of Sociology, University of Oxford, chapter 13, pp. 213-237.
23. Jøsang, A., Gray, E., Kinateder, M.: Analysing Topologies of Transitive Trust. In Dimitrakos, T., Martinelli, F., eds.: Proceedings of the First International Workshop on Formal Aspects in Security & Trust (FAST2003), Pisa, Italy (September 2003) 9–22