A Method to Evaluate Uncertain and Conflicting Trust and Authenticity Statements

Andreas Gutscher

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

Abstract. Countless Internet applications and platforms make it easy to communicate, to collaborate and to exchange information and opinions with a huge number of individuals. However, it is getting more and more difficult to distinguish honest and reliable individuals from malicious users distributing false information or malware. Digital signatures, webs of trust and reputation systems can help to securely identify communication partners and to estimate the trustworthiness of others, but there is a lack of trust and authenticity evaluation methods that do not show counterintuitive effects in the case of conflicting opinions.
This article proposes a new integrated method to evaluate uncertain and conflicting trust and authenticity statements. It introduces a set of operators and inference rules for combining and reasoning with these statements, it defines an approach to compute the resulting confidence values of derived statements and it compares different computation algorithms. The computation is based on a probability theoretical model in order to exclude inconsistencies and counter-intuitive effects.

1 Introduction

An increasing number of different Internet applications, platforms and social networks makes it easy to communicate with a huge number of individuals, to exchange and share information, news, photos, files and product recommendations and to socialize with people sharing similar interests. However, for users participating in a large number of social networks, discussion forums, etc. it is getting more and more difficult to find out who their new "friends" actually are and whether they can trust them.

With a *reputation system* users can share their knowledge and opinions about other users. The reputation system collects and evaluates the opinions of all users about the trustworthiness of others. On request it computes the resulting confidence value for the requested entity according to a *trust model*.

* D. Chadwick, I. You and H. Chang (Eds.): Proceedings of the 1st International Workshop on Managing Insider Security Threats (MIST2009), Purdue University, West Lafayette, USA, June 16, 2009.

The authenticity of all opinion statements should be protected, e. g., with digital signatures, to prevent manipulations and to make the evaluation verifiable to the users. Digital signatures are only useful if the users can identify the signature key holder. If no global trusted public key infrastructure is available, users can share their knowledge about the ownership of keys in a so-called *web of trust* (e. g., the PGP/GnuPG *web of trust* [1]) by exchanging digitally signed identity certificates. However, these authenticity statements are only useful if the users can verify that the issuer is trustworthy to verify the ownership of public keys. Trust and authenticity evaluation are thus highly interdependent.

Various different computational trust models, reputation systems and applications using trust have been proposed [1–10]. To obtain an intuitive and consistent trust model one must define clearly what a confidence value represents and find a sound mathematical basis for the computation with confidence values. Confidence values have strong similarities to probability values. The most sophisticated trust models are therefore based on probability theory. Maurer [4] proposed a probabilistic model in which confidence values for trust and authenticity relations correspond to probability values. However, it does not model negative opinions (distrust) and entities cannot have more than one key. Credential Networks and related models proposed by Haenni [6], Jonczy [11] and Kohlas [10] also model confidence values as probabilities. Besides degrees of support and uncertainty the confidence values can express also degrees of refutation (e. g., distrust). However, confidence values may contain either degrees of belief and ignorance[1], disbelief and ignorance, or belief and disbelief, but they cannot contain degrees of belief, disbelief and ignorance at the same time. Jøsang's Subjective Logic [5] can express opinions with degrees of belief, ignorance and disbelief (at the same time), but the approach to compute the resulting confidence values is (although based on probability theory, too) quite different. In the model of Maurer and in Credential Networks the initial confidence values are uncertain and the inference rules are deterministic, whereas in Subjective Logic the uncertainty is modeled in the operators of the inference rules, i. e., the confidence value of a conclusion is computed from the confidence values of the preconditions. Unfortunately, this leads to the problem that the resulting confidence value generally depends on the order in which the operators are applied. It seems that Subjective Logic can not be used to evaluate arbitrary networks of trust and authenticity statements without using questionable workarounds [12].

If users can express both positive (supporting) and negative (refuting) opinions, then the combination of contradictory opinions can lead to *conflicts*. In Credential Networks and Subjective Logic the probability mass associated with conflicting combinations is eliminated and the remaining probability mass is renormalized. Zadeh [13] has shown that conflict elimination and re-normalization approaches (like Dempster's rule of combination [14]) can produce counterintuitive effects.

This article proposes a new integrated approach to evaluate uncertain and conflicting trust and authenticity statements without eliminating conflict. This

[1] the terms *uncertainty* and *ignorance* are used interchangeable

avoids the counter-intuitive effects of re-normalizations. The trust model is based on a combination of the inference rules from [15] and the calculus and operators from [16], extended by a new operator for reasoning with authenticity statements. Sect. 2 describes the representation of trust and authenticity statements, Sect. 3 the representation of confidence values and the corresponding operators. The new inference rules are formulated in Sect. 4. Sect. 5 proposes different algorithms to compute the resulting confidence value, Sect. 6 compares the computation time of these algorithms and Sect. 7 concludes the article.

2 Model of Trust and Authenticity Statements

An *opinion* refers to a trust or authenticity statement H_j with an associated confidence value t_j. A *first-hand* opinion is an opinion that is based only on the experience and knowledge of a *single* entity (the *trustor* or *issuer*) and that is *independent* of other opinions. A *second-hand* opinion is an opinion that is derived from other opinions and that is thus not independent.

We define *trust* as *"a unidirectional relation between a trustor and a trustee expressing the strong belief of the trustor that the trustee will behave as expected with respect to a particular capability within a particular context"* [15]. Therefore we represent the standard form of a trust statement as follows:

$$\mathrm{Trust}(\mathit{trustor}, \mathit{trustee}, r, h_{\min}..h_{\max}) \tag{1}$$

The *trustor* can be an entity or a key, the *trustee* an entity, a description or a key (see Tab. 1). An *entity* (E_A, E_B, ...) can be a person, an organization, a network node, etc. referred to by a *local* identifier. To exchange opinions with others users have to use unique *descriptions* or *public keys* to refer to other entities. A *description* (D_A, D_B, ...) consists of a list of names, identifiers or attributes that uniquely identifies the described entity. Entities may have several different descriptions. A *public key* (K_A, K_B, ...) is the public part of an asymmetric key pair. The holder uses the key pair to sign trust or authenticity statements (certificates). An entity can use several different key pairs at the same time.

Table 1. Trust and authenticity statements (relations and certificates)

	Trust statements		Authenticity
	Standard form	Internal form	statements
Relation	$\mathrm{Trust}(E_A, E_B, r, h_{\min}..h_{\max})$	$\mathrm{Trust}(E_A, E_B, r, h, l)$	$\mathrm{Auth}(E_A, K_B, E_B)$
	$\mathrm{Trust}(E_A, K_B, r, h_{\min}..h_{\max})$	$\mathrm{Trust}(E_A, K_B, r, h, l)$	$\mathrm{Auth}(E_A, D_B, E_B)$
	$\mathrm{Trust}(E_A, D_B, r, h_{\min}..h_{\max})$	$\mathrm{Trust}(E_A, D_B, r, h, l)$	$\mathrm{Auth}(E_A, K_B, D_B)$
Certificate	$\mathrm{Trust}(K_A, K_B, r, h_{\min}..h_{\max})$	$\mathrm{Trust}(K_A, K_B, r, h, l)$	$\mathrm{Auth}(K_A, K_B, D_B)$
	$\mathrm{Trust}(K_A, D_B, r, h_{\min}..h_{\max})$	$\mathrm{Trust}(K_A, D_B, r, h, l)$	

The capability r refers to an application specific capability (r_1, r_2, ...) or to the capability r_{PKI}, which represents the capability to honestly and carefully verify that a description uniquely refers to the holder of a particular key pair.

We distinguish different types of trust identified by a different number of recommendation hops (h): *Functional trust* expresses the belief that the trustee *has* the capability r and is described by $h = 0$. *Recommendation trust* for $h = 1$ hop expresses the belief that the trustee can *recommend* someone with capability r, *recommendation trust* for $h = 2$ hops that the trustee can *recommend someone who can recommend* someone with capability r, etc. Each standard form trust statement can specify the desired range of recommendation hops $h_{\min}..h_{\max}$.

For the evaluation of trust statements we need in addition trust statements in the slightly different *internal form*. These trust statements refer not to a range, but to a single recommendation hop value $h \geq 0$ and they have an additional parameter, the *chain length* $l \geq 1$:

$$\text{Trust}(\mathit{trustor}, \mathit{trustee}, r, h, l) \tag{2}$$

Trust is not transitive in general, but trust statements can be combined in certain cases to trust chains according to the transitive trust inference rule (7) described in Sect. 4.1. The chain length l of the derived trust statement refers to the number of first-hand trust statements in the trust chain.

Authenticity statements express the strong belief of the issuer that a description belongs to an entity, that a public key belongs to an entity or that a description belongs to the holder of a public key:

$$\text{Auth}(\mathit{issuer}, \mathit{actor}_1, \mathit{actor}_2) \tag{3}$$

The *issuer* is an entity or a public key, actor_1 and actor_2 are entities, descriptions or public keys. All four possible combinations are listed in Tab. 1[2].

3 Confidence Values

This section introduces discrete and continuous confidence values as well as operators for reasoning with discrete confidence values. Users express their opinions with continuous confidence values while the discrete confidence values are used internally only for reasoning with opinions.

3.1 Representation of Discrete and Continuous Confidence Values

Users can have different and possibly conflicting opinions about trust and authenticity statements. Therefore, we can not definitively decide whether a statement H is "true" or "false". We can only evaluate known indications that *support* or *refute* H. It is possible that neither supporting nor refuting or that both supporting and refuting indications for H are found. Therefore we describe knowledge of supporting and refuting indications *independently*. For each statement H we introduce the *propositions* H^+ and H^- to describe that the reputation

[2] certificates can not contain local identifiers for entities ($E_A, E_B, \ldots$) because they would be meaningless to other entities

system is aware of indications that imply that H must be true and that H must be false, respectively. We also introduce the four *discrete* confidence values *belief* $(+)$, *ignorance* $(\varnothing)$, *disbelief* $(-)$ and *conflict* $(\pm)$ to represent the four possible combinations of these propositions (see Tab. 2). They can be seen as "truth values" of a paraconsistent logic [16].

Table 2. Discrete confidence values

Propositions	Discrete confidence value	Semantics
$\{H^+\}$	$t' = +$ (*belief*)	"the indications imply that H must be true"
$\{\}$	$t' = \varnothing$ (*ignorance*)	"there are no relevant indications about H"
$\{H^-\}$	$t' = -$ (*disbelief*)	"the indications imply that H must be false"
$\{H^+, H^-\}$	$t' = \pm$ (*conflict*)	"the indications imply that H must be true and that H must be false at the same time"

As statements can in fact not be both true and false at the same time we can conclude that *first-hand* opinions can not have the confidence value *conflict*. However, if we combine statements of *different* (disagreeing) entities, it is possible to find both H^+ and H^-, i. e., the confidence value of derived (*second-hand*) opinions can be *conflict*. Conflict must not be confused with partial support and partial refutation (*ambivalent opinions*). An entity that has for example experienced some positive and some negative interactions can express this opinion with *continuous* confidence values.

Continuous confidence values $t = (b, i, d, c)$ with $b, i, d, c \in [0, 1]$ and $b + i + d + c = 1$ express *degrees* of belief, ignorance, disbelief and conflict. The value b represents the issuer's subjective estimation of the probability that there are indications supporting (but no refuting) H. Similarly, d represents the subjective estimation of the probability that there are indications refuting (but no supporting) H. c represents the subjective estimation of the probability that there are both supporting and refuting indications for H at the same time, and i represents the subjective estimation of the probability that there are neither supporting nor refuting indications for H. For the same reason as before, c must be zero in all first-hand opinions, whereas second-hand opinions can contain conflict. Nevertheless, ambivalent first-hand opinions can be expressed by continuous confidence values with both $b > 0$ and $d > 0$. A user that has made many positive and few negative experiences can choose, for example, a first-hand confidence value with $b = 0.7$ and $d = 0.1$ (i. e., $t = (0.7, 0.2, 0.1, 0)$. Thus, in first-hand statements b can be seen as the lower bound and $1 - d$ as the upper bound for the estimated subjective probability that H must be true.

The degrees of ignorance and conflict in resulting confidence values have different meanings, and applications should handle high degrees of ignorance and conflict differently: A high degree of ignorance indicates that the reputation system has little information about the requested statement and suggests searching more relevant statements, if possible. A high degree of conflict, however, shows that the requested statement H is controversial. This suggests that the requester

should verify whether the trust and authenticity assignments he made and that cause the conflict are correct.

Continuous confidence value can be condensed to a single value w, if desired:

$$w = b + w_i i + w_d d + w_c c \tag{4}$$

The parameters w_i, w_d and w_c represent weights for the degrees of ignorance, disbelief and conflict, e. g., $w_i = 0.5$, $w_d = -1$ and $w_c = 0$. They can be chosen according to the preferences of the application and allow for rather optimistic or rather pessimistic behavior in the cases of uncertainty and conflict.

3.2 Operators to Combine Discrete Confidence Values

This section describes the recommendation and authentication operators. The operators define whether H_z^+ and H_z^- can be derived from a set of premises (H_x^+, H_x^-, H_y^+, H_y^-). The short notation with statements is provided for convenience and will be used to formulate the inference rules in Sect. 4.

Recommendation Operator The recommendation operator ($\otimes$) is used to concatenate two trust statements or a trust with an authenticity statement. It is reasonable for a user to adopt the opinions of trustworthy entities. However, it is not reasonable (it is in fact even dangerous) to assume that untrustworthy (malicious or incompetent) entities always tell the opposite of the truth. Instead, opinions of untrustworthy entities should be ignored. Therefore, we do not draw any conclusions from H_x^-. The operator is thus defined as follows:

$$\frac{H_x \otimes H_y}{H_z} \Leftrightarrow \frac{H_x^+ \quad H_y^+}{H_z^+}, \frac{H_x^+ \quad H_y^-}{H_z^-} \tag{5}$$

This reads as follows: H_z follows from a combination of H_x and H_y with the recommendation operator. If there are supporting indications for H_x and for H_y, then infer H_z^+. If there are supporting indications for H_x and refuting indications for H_y, then infer H_z^-. Fig. 1 (left) shows the corresponding "truth table".

$t'_z = t'_x \otimes t'_y$		t'_x			
		$+$	$\varnothing$	$-$	$\pm$
t'_y	$+$	$+$	$\varnothing$	$\varnothing$	$+$
	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
	$-$	$-$	$\varnothing$	$\varnothing$	$-$
	$\pm$	$\pm$	$\varnothing$	$\varnothing$	$\pm$

$\odot$	$+$	$\varnothing$	$-$	$\pm$
$+$	$+$	$\varnothing$	$-$	$\pm$
$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$	$\varnothing$
$-$	$-$	$\varnothing$	$\varnothing$	$-$
$\pm$	$\pm$	$\varnothing$	$-$	$\pm$

Fig. 1. Recommendation and authentication operator truth tables

Authentication Operator The authentication operator ($\odot$) is used to reason with two authenticity relations between entities, descriptions and public keys:

$$\frac{H_x \odot H_y}{H_z} \Leftrightarrow \frac{H_x^+ \quad H_y^+}{H_z^+}, \frac{H_x^+ \quad H_y^-}{H_z^-}, \frac{H_x^- \quad H_y^+}{H_z^-} \tag{6}$$

The operator definition can be understood as follows: Assume H_x and H_y represent statements like "A and B belong together" and "B and C belong together", respectively. If we have supporting indications for both statements, then this supports that A and C belong together (H_z). If we have indications that A and B belong together but that B does not belong to C, then we conclude that A does not belong to C either. If neither A belongs to B nor does B belong to C, then we can draw no conclusion about A and C. Fig. 1 (right) shows the corresponding truth table.

4 Inference Rules

The inference rules specify which conclusions the reputation system can draw from a set of given trust and authenticity propositions.

4.1 Transitive Trust Inference Rule

This inference rule describes the *transitivity* property of trust statements. It defines in which cases two trust statements for the same capability r can be combined with the recommendation operator in order to derive a new trust statement from the trustor of the first statement (A) to the trustee of the second statement (C). The trustor A can be an entity (E_A) or a public key (K_A). The second statement can be a trust statement 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).

$$\frac{\text{Trust}(A, B, r, h + l_2, l_1) \otimes \text{Trust}(B, C, r, h, l_2)}{\text{Trust}(A, C, r, h, l_1 + l_2)} \tag{7}$$

This inference rule differs from other proposed transitive trust inference rules in that it allows the combination of trust statements only if the number of recommendation hops matches: The number of recommendation hops of the first statement must equal the sum of the recommendation hops plus the chain length of the second statement. The chain length of the resulting statement is the sum of the chain lengths of the input statements. This ensures that the recommendation hop value of the trust statements decreases by one throughout the chain of first-hand trust relations (e. g., $h = 2$, $h = 1$, $h = 0$).

The example in Fig. 2 illustrates the inference rule. The transitive trust inference rule allows to combine $H_1^+ = \text{Trust}^+(E_A, E_B, r, 2, 1)$ with $H_2^+ = \text{Trust}^+(E_B, E_C, r, 1, 1)$ to $H_4^+ = \text{Trust}^+(E_A, E_C, r, 1, 2)$ and then H_4^+ with $H_3^- = \text{Trust}^-(E_C, E_D, r, 0, 1)$ to $H_5^- = \text{Trust}^-(E_A, E_D, r, 0, 3)$.

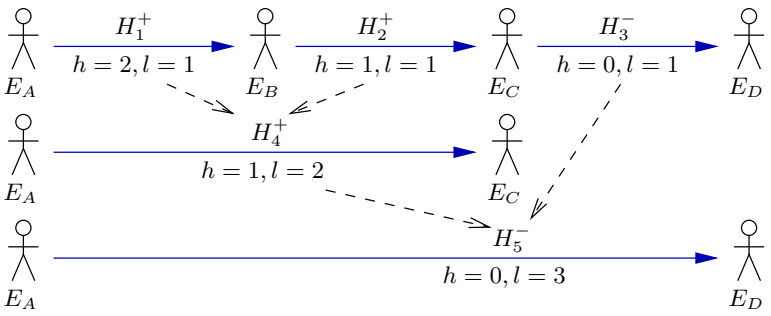


Fig. 2. Example for application of the transitive trust inference rule

4.2 Trust in Entities, Keys and Descriptions

A number of simple rules allow to infer from trust assigned to an entity to trust assigned to the holder of a key and to trust assigned to an entity identified by a description, and vice versa. If an entity is trustworthy, then the holder of a key that belongs to this entity is trustworthy, too, and vice versa:

$$\frac{\text{Auth}(E_A, K_C, E_C) \otimes \text{Trust}(E_A, E_C, r, h, l)}{\text{Trust}(E_A, K_C, r, h, l)} \tag{8}$$

$$\frac{\text{Auth}(E_A, K_C, E_C) \otimes \text{Trust}(E_A, K_C, r, h, l)}{\text{Trust}(E_A, E_C, r, h, l)} \tag{9}$$

If an entity is trustworthy, then the entity identified by a description that belongs to this entity is trustworthy, too, and vice versa:

$$\frac{\text{Auth}(E_A, D_C, E_C) \otimes \text{Trust}(E_A, E_C, r, h, l)}{\text{Trust}(E_A, D_C, r, h, l)} \tag{10}$$

$$\frac{\text{Auth}(E_A, D_C, E_C) \otimes \text{Trust}(E_A, D_C, r, h, l)}{\text{Trust}(E_A, E_C, r, h, l)} \tag{11}$$

If the holder of a key is trustworthy, then the entity identified by a description that belongs to this key holder is trustworthy, too, and vice versa. This applies to trust relations and trust certificates:

$$\frac{\text{Auth}(E_A, K_C, D_C) \otimes \text{Trust}(E_A, K_C, r, h, l)}{\text{Trust}(E_A, D_C, r, h, l)} \tag{12}$$

$$\frac{\text{Auth}(E_A, K_C, D_C) \otimes \text{Trust}(E_A, D_C, r, h, l)}{\text{Trust}(E_A, K_C, r, h, l)} \tag{13}$$

$$\frac{\text{Auth}(K_A, K_C, D_C) \otimes \text{Trust}(K_A, K_C, r, h, l)}{\text{Trust}(K_A, D_C, r, h, l)} \tag{14}$$

$$\frac{\text{Auth}(K_A, K_C, D_C) \otimes \text{Trust}(K_A, D_C, r, h, l)}{\text{Trust}(K_A, K_C, r, h, l)} \tag{15}$$

4.3 Local Authenticity Inference Rule

If an entity E_A has *partial* knowledge about whether an entity E_B is the holder of a key K_B, whether a description D_B refers to the entity E_B or whether the description D_B refers to the holder of the key K_B, then it can draw further conclusions about the confidence values of the authenticity statements between E_B, K_B and D_B. If the confidence values of two corresponding authenticity relations are known, then the confidence value of the third authenticity relation can be derived with the authentication operator:

$$\frac{\mathrm{Auth}(E_A, K_C, D_C) \odot \mathrm{Auth}(E_A, K_C, E_C)}{\mathrm{Auth}(E_A, D_C, E_C)} \tag{16}$$

$$\frac{\mathrm{Auth}(E_A, K_C, D_C) \odot \mathrm{Auth}(E_A, D_C, E_C)}{\mathrm{Auth}(E_A, K_C, E_C)} \tag{17}$$

$$\frac{\mathrm{Auth}(E_A, K_C, E_C) \odot \mathrm{Auth}(E_A, D_C, E_C)}{\mathrm{Auth}(E_A, K_C, D_C)} \tag{18}$$

4.4 Authenticity Inference with Authenticity Confirmation

If a trustor (E_A or K_A) trusts a trustee (E_B or K_B) to issue only correct authenticity relations or identity certificates (property r_{PKI}), then the trustor can conclude that authenticity relations or identity certificates of the trustee are correct:

$$\frac{\mathrm{Trust}(E_A, E_B, r_{\mathrm{PKI}}, 0, l) \otimes \mathrm{Auth}(E_B, K_C, D_C)}{\mathrm{Auth}(E_A, K_C, D_C)} \tag{19}$$

$$\frac{\mathrm{Trust}(E_A, K_B, r_{\mathrm{PKI}}, 0, l) \otimes \mathrm{Auth}(K_B, K_C, D_C)}{\mathrm{Auth}(E_A, K_C, D_C)} \tag{20}$$

$$\frac{\mathrm{Trust}(K_A, K_B, r_{\mathrm{PKI}}, 0, l) \otimes \mathrm{Auth}(K_B, K_C, D_C)}{\mathrm{Auth}(K_A, K_C, D_C)} \tag{21}$$

4.5 Uniqueness Conditions

Two further conclusions can be drawn from the condition that each public key has only one holder and that each description refers to only one entity. If A knows that E_B is the holder of K_B, then it can infer that all other entities are not the holder of K_B. Similarly, if A knows that E_B has the description D_B, then it can infer that all other entities do not have the description D_B (A can be an entity or a key).

$$\frac{Auth^{+}(A, K_B, E_B)}{Auth^{-}(A, K_B, E_j)}, \frac{Auth^{+}(A, D_B, E_B)}{Auth^{-}(A, D_B, E_j)} \qquad \forall E_j \neq E_B \tag{22}$$

5 Confidence Value Computation

The reputation system collects all issued first-hand trust and authenticity opinions H_j with associated continuous confidence value t_j (with $c_j = 0$). Users can then send requests in the form of a standard form trust statement or an authenticity statement to the reputation system. The reputation system then processes all collected opinions. It applies the inference rules to derive trust and authenticity statements and it computes the resulting continuous confidence value t_0 of the requested statement H_0 from the confidence values of the relevant first-hand statements. As the components of the continuous first-hand confidence values (b, i and d) represent probabilities, we define the resulting confidence value by a random experiment and propose different algorithms for the computation of the resulting confidence value.

5.1 Probabilistic Model for the Confidence Value Computation

The components of the computed *resulting* confidence value $t_0 = (b_0, i_0, d_0, c_0)$ for H_0 are computed from the combination of all available first-hand opinions with the inference rules under the assumption that the confidence values of the opinions of the requestor are correct. In short, b_0 is the computed lower bound for the probability that the combination of the available first-hand opinions leads to the conclusion that H_0 must be true (but not that H_0 must be false). Similarly, d_0 is the computed lower bound for the probability that the combination of the available first-hand opinions leads to the conclusion that H_0 must be false (but not that H_0 must be true). The degree of conflict c_0 is the computed probability that the combination of the first-hand opinions leads to the contradicting conclusion that H_0 must be both true *and* false at the same time. The degree of ignorance is the remaining probability $i_0 = 1 - b_0 - d_0 - c_0$.

The following description of a random experiment provides a more detailed definition for t_0: We assume that the reputation system has collected J first-hand opinions, i. e., the statements H_j ($j = 1, 2, \ldots J$) with associated continuous confidence values $t_j = (b_j, i_j, d_j, 0)$. For each first-hand statement H_j choose a *discrete* confidence value t'_j from $\{+, \varnothing, -\}$ according to the weights b_j, i_j and d_j, i. e., choose $t'_j = +$ with probability b_j, $t'_j = \varnothing$ with probability i_j and $t'_j = -$ with probability d_j. Statements with the discrete confidence value *ignorance* don't contribute knowledge and can be discarded[3]. Each remaining first-hand statement H_j with associated discrete confidence value t'_j corresponds to a set of first-hand propositions according to Tab. 2.

The inference rules always operate on trust propositions in the *internal* representation. We therefore have to replace each standard-form trust statement $\mathrm{Trust}(A, B, r, h_{\min}..h_{\max})$ by a list of single-hop trust statements in *internal form* with chain length $l = 1$: $\mathrm{Trust}(A, B, r, h_{\min}, l)$, $\mathrm{Trust}(A, B, r, h_{\min} + 1, l)$, $\ldots$, $\mathrm{Trust}(A, B, r, h_{\max}, l)$. The internal trust statements inherit their assigned

[3] this optimization does not change the resulting confidence value, the resulting continuous confidence value t_0 nevertheless contains the correct degree of ignorance

discrete confidence value from the standard-form trust statement. Next, we apply all inference rules (see Sect. 4) to derive all (positive and negative) deducible propositions from the set of all known first-hand propositions and all already derived propositions. To get back to trust statements in standard form we conclude $H_0^+ = \mathrm{Trust}^+(A, B, r, h_{\min}..h_{\max})$ if we have been able to derive a proposition $H_{0,h}^+ = \mathrm{Trust}^+(A, B, r, h, l)$ with $h_{\min} \leq h \leq h_{\max}$. Similarly, we conclude H_0^- if we have been able to derive a proposition $H_{0,h}^-$.

To obtain the resulting continuous confidence value of a requested trust or authenticity statement we compute the probability that the random experiment leads to a set of first-hand propositions from which we can derive positive and negative propositions for the requested statement H_0. The components of the resulting confidence value $t_0 = (b_0, i_0, d_0, c_0)$ are defined as follows: b_0 is the probability that H_0^+ (but not H_0^-) can be derived and d_0 is the probability that H_0^- (but not H_0^+) can be derived. The probability that neither H_0^+ nor H_0^- can be derived is i_0, and c_0 is the probability that both H_0^+ and H_0^- can be derived.

In contrast to other trust models (e. g., [5, 6, 10, 11]) we propose *not to eliminate* the degree of conflict, not only to avoid counter-intuitive effects of renormalizations but also because it provides valuable information to the requesting user or application (see Sect. 3.1).

5.2 Approximation and Exact Computation Algorithms

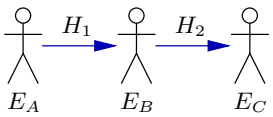
This section presents different possibilities to implement the computation of an approximation or of the exact value of the resulting continuous confidence value t_0 according to Sect. 5.1. All exact algorithms return the same resulting confidence value t_0, but differ in computation time. The result of the approximation gets arbitrarily close to the exact result if the number of iterations is sufficiently large.

To keep the computation time small we recommend for all algorithms to *precompute all possible paths*: We first set up a "superposition" of possible first-hand propositions. For each statement H_j with continuous confidence value $t_j = (b_j, i_j, d_j, 0)$ we select H_j^+ if $b_j > 0$ and we select (possibly in addition) H_j^- if $d_j > 0$. Then we translate all trust propositions into the internal form, apply all inference rules and record the dependencies, i. e., we trace which sets of first-hand propositions (premises) allow to derive which conclusions. Each set of first-hand propositions that allows to (directly or indirectly) derive the positive requested proposition H_0^+ is called a *positive path* for H_0, each set that allows to derive the negative proposition H_0^- a *negative path* for H_0. Next, we select the set of positive paths and the set of negative paths for H_0 and minimize these paths, i. e., we remove all paths that contain at least one other path in the set. We finally obtain the set of minimal positive paths $A^+ = \{a_1^+, a_2^+, \ldots a_{k^+}^+\}$ and the set of minimal negative paths $A^- = \{a_1^-, a_2^-, \ldots a_{k^-}^-\}$.

Approximation with Monte-Carlo Simulation An obvious approach to determine an *approximation* for the resulting confidence value is to run the

described random experiment N times and to count in how many experiments the selected set of first-hand propositions contains at least one positive (but no negative) path (n_b), no paths (n_i), at least one negative (but no positive) path (n_d) or both positive and negative paths (n_c). The approximation for the confidence value is $\bar{t}_0 = \frac{1}{N}(n_b, n_i, n_d, n_c)$. The choice of N allows to adjust the trade-off between precision and computation time.

Possible Worlds Algorithm An simple algorithm to compute the exact value is to go through the list of all possible combinations of first-hand propositions (so-called *possible worlds*), to compute the probability of each of those possible worlds and to check for each world whether the set of first-hand propositions of this world contains the minimal paths. The sum of all probabilities of all worlds that contain at least one positive and at least one negative path is c_0, b_0 is the sum of probabilities of all worlds that contain at least one positive, but no negative path, and d_0 the sum of probabilities of all worlds that contain at least one negative, but no positive path. The degree of ignorance is $i_0 = 1 - b_0 - d_0 - c_0$.



j	H_j	b_j	i_j	d_j
1	Trust$(E_A, E_B, r_1, 1)$	0.5	0.15	0.35
2	Trust$(E_B, E_C, r_1, 0)$	0.7	0.1	0.2

$H_{1,1}$	$H_{2,0}$	Probability	H_0
$H^+_{1,1}$	$H^+_{2,0}$	$b_1 b_2$	H^+_0
$H^+_{1,1}$		$b_1 i_2$	
$H^+_{1,1}$	$H^-_{2,0}$	$b_1 d_2$	H^-_0
	$H^+_{2,0}$	$i_1 b_2$	
		$i_1 i_2$	
	$H^-_{2,0}$	$i_1 d_2$	
$H^-_{1,1}$	$H^+_{2,0}$	$d_1 b_2$	
$H^-_{1,1}$		$d_1 i_2$	
$H^-_{1,1}$	$H^-_{2,0}$	$d_1 d_2$	

Fig. 3. Scenario and possible worlds table of Example 1

Example 1: Simple Trust Chain with Possible Worlds Algorithm The example scenario in Fig. 3 (left) consists of two trust statements: H_1 is a recommendation trust statement for one recommendation hop ($h = 1$), and H_2 is a functional trust statement ($h = 0$). E_A wants to compute the resulting functional trustworthiness of E_C ($H_0 = \text{Trust}(E_A, E_C, r_1, 0)$).

First, the trust statements in standard form have to be replaced by corresponding trust statements in internal form: H_1 by $H_{1,1} = \text{Trust}(E_A, E_B, r_1, 1, 1)$ and H_2 by $H_{2,0} = \text{Trust}(E_B, E_C, r_1, 0, 1)$. Both refer to the same property r_1, it is therefore possible to combine $H_{1,1}$ and $H_{2,0}$ with the transitive trust inference rule (7) to the new functional trust statement $H_{0,0} = \text{Trust}(E_A, E_C, r_1, 0, 2)$: $H^+_{1,1}, H^+_{2,0} \vdash H^+_{0,0}$ ($H^+_{1,1}$ and $H^+_{2,0}$ allow to drive $H^+_{0,0}$) and $H^+_{1,1}, H^-_{2,0} \vdash H^-_{0,0}$. Thus, there is only one positive path $a^+_1 = \{H^+_{1,1}, H^+_{2,0}\}$ and one negative path $a^-_1 = \{H^+_{1,1}, H^-_{2,0}\}$ for $H_{0,0}$ and thus for H_0.

Fig. 3 (right) shows all possible combinations of the first-hand propositions, the probability that this world occurs and the propositions that can be derived in this world. There are no worlds in which both H_0^+ and H_0^- can be derived, thus $c_0 = 0$. H_0^+ can be derived only in the first world, therefore $b_0 = b_1 b_2$. Similarly, H_0^- can be derived only in the third world, therefore $d_0 = b_1 d_2$. The degree of ignorance is the remaining probability mass $i_0 = 1 - b_0 - d_0 - c_0$. With the values in Fig. 3 we obtain $t_0 = (0.35, 0.55, 0.1, 0)$.

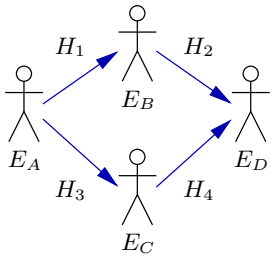
Grouped Possible Worlds Algorithm The possible worlds algorithm can be improved by subdividing the set of relevant first-hand statements into as few groups $g_1, \ldots g_u$ as possible. Two statements, H_j and H_m, belong to the same group if and only if the following condition holds for each positive, negative and conflicting path: If the path contains a proposition for H_j (H_j^+ or H_j^-), then it must also contain a proposition for H_m (H_m^+ or H_m^-).

In the preparation step we construct for each group a list of all relevant combinations of propositions of the statements in the group. This list contains all combinations that contain exactly one proposition (i. e., either H_j^+ or H_j^-)[4]for each statement and that is identical to the corresponding section of at least one (positive or negative) path. An additional element of this list consists of an empty set. It represents all remaining possible combinations of propositions, i. e., all combinations that contain neither H_j^+ nor H_j^- for at least one statement H_j of the group. We can subsume these combinations because the have the same effect on the derivability of propositions of H_0. For each element of the list we compute the probability that this combination will occur (within the group) from the continuous confidence values of the statements. The probability associated with the empty set is the sum of the probabilities of the contained combinations of propositions (i. e., the remaining probability). Thus, the sum of all probabilities is one.

Next, in the main step, we go through all possible worlds. Each world consists of one possible combination of these prepared proposition-combinations of the groups, i. e., for each groups we select one proposition-combination from the prepared list of the group. We multiply the precomputed probabilities of the chosen proposition-combinations to obtain the resulting probability of the world. Finally, we compute b_0, i_0, d_0 and c_0 just as in the possible worlds algorithm.

Example 2: Parallel Trust Chain with Grouped Possible Worlds Algorithm The scenario in Fig. 4 consists of two parallel trust chains from E_A to E_D. E_A requests the confidence value for the resulting functional trustworthiness of E_D ($H_0 = \text{Trust}(E_A, E_D, r_1, 0)$). The trust statements in standard form are replaced by statements in internal form: H_1 by $H_{1,1} = \text{Trust}(E_A, E_B, r_1, 1, 1)$, H_2 by $H_{2,0} = \text{Trust}(E_B, E_D, r_1, 0, 1)$, H_3 by $H_{3,1} = \text{Trust}(E_A, E_C, r_1, 1, 1)$ and H_4 by $H_{4,0} = \text{Trust}(E_C, E_D, r_1, 0, 1)$. We can combine $H_{1,1}$ with $H_{2,0}$ and $H_{3,1}$ with $H_{4,0}$ with the transitive trust inference rule (7). We obtain

[4] no combination can contain both H_j^+ and H_j^- because $c_j = 0$



j	H_j	b_j	i_j	d_j
1	Trust$(E_A, E_B, r_1, 1)$	0.8	0.15	0.05
2	Trust$(E_B, E_D, r_1, 0)$	0.7	0.1	0.2
3	Trust$(E_A, E_C, r_1, 1)$	0.9	0.1	0
4	Trust$(E_C, E_D, r_1, 0)$	0.8	0.1	0.1

Fig. 4. Scenario of Example 2

two positive paths $A^+ = \{\{H^+_{1,1}, H^+_{2,0}\}, \{H^+_{3,1}, H^+_{4,0}\}\}$ and two negative paths $A^- = \{\{H^+_{1,1}, H^-_{2,0}\}, \{H^+_{3,1}, H^-_{4,0}\}\}$. Propositions for $H_{1,1}$ and $H_{2,0}$ appear always together in paths, the same holds for $H_{3,1}$ and $H_{4,0}$. Therefore we can divide the statements into two groups $g_1 = \{H_{1,1}, H_{2,0}\}$ and $g_1 = \{H_{3,1}, H_{4,0}\}$.

In the preparation step we set up a list for each group that contains all relevant possible combinations of the propositions and their probabilities (see Tab. 3). For each group we find three relevant combinations: one combination supports a positive path and one a negative path. The third entry with the empty set represents the remaining combinations.

Table 3. Preparation step for the groups in Example 2

Propositions g_1	Probability
$\{H^+_{1,1}, H^+_{2,0}\}$	$b_1 b_2$
$\{H^+_{1,1}, H^-_{2,0}\}$	$b_1 d_2$
$\{\}$	$1 - b_1 b_2 - b_1 d_2$

Propositions g_2	Probability
$\{H^+_{3,1}, H^+_{4,0}\}$	$b_3 b_4$
$\{H^+_{3,1}, H^-_{4,0}\}$	$b_3 d_4$
$\{\}$	$1 - b_3 b_4 - b_3 d_4$

Table 4. Confidence value computation in the parallel trust chain example

Table 5. Confidence value computation in Example 2

g_1	g_2	Probability	H_0
$\{H^+_{1,1}, H^+_{2,0}\}$	$\{H^+_{3,1}, H^+_{4,0}\}$	$b_1 b_2 b_3 b_4$	H^+_0
$\{H^+_{1,1}, H^+_{2,0}\}$	$\{H^+_{3,1}, H^-_{4,0}\}$	$b_1 b_2 b_3 d_4$	H^+_0, H^-_0
$\{H^+_{1,1}, H^+_{2,0}\}$	$\{\}$	$b_1 b_2 (1 - b_3 b_4 - b_3 d_4)$	H^+_0
$\{H^+_{1,1}, H^-_{2,0}\}$	$\{H^+_{3,1}, H^+_{4,0}\}$	$b_1 d_2 b_3 b_4$	H^+_0, H^-_0
$\{H^+_{1,1}, H^-_{2,0}\}$	$\{H^+_{3,1}, H^-_{4,0}\}$	$b_1 d_2 b_3 d_4$	H^-_0
$\{H^+_{1,1}, H^-_{2,0}\}$	$\{\}$	$b_1 d_2 (1 - b_3 b_4 - b_3 d_4)$	H^-_0
$\{\}$	$\{H^+_{3,1}, H^+_{4,0}\}$	$(1 - b_1 b_2 - b_1 d_2) b_3 b_4$	H^+_0
$\{\}$	$\{H^+_{3,1}, H^-_{4,0}\}$	$(1 - b_1 b_2 - b_1 d_2) b_3 d_4$	H^-_0
$\{\}$	$\{\}$	$(1 - b_1 b_2 - b_1 d_2)(1 - b_3 b_4 - b_3 d_4)$	

To compute the resulting confidence value t_0 for H_0 we set up Tab. 5 with all $3 \cdot 3 = 9$ possible combinations of the entries in the lists (possible worlds), the probabilities of each world and the derivable propositions for H_0. Then we add the probabilities and obtain $b_0 = b_1b_2b_3b_4 + b_1b_2(1 - b_3b_4 - b_3d_4) + (1 - b_1b_2 - b_1d_2)b_3b_4$, $i_0 = (1-b_1b_2-b_1d_2)(1-b_3b_4-b_3d_4)$, $d_0 = b_1d_2b_3d_4+b_1d_2(1-b_3b_4-b_3d_4) + (1 - b_1b_2 - b_1d_2)b_3d_4$ and $c_0 = b_1b_2b_3d_4 + b_1d_2b_3b_4$. With the values in Fig. 4 we obtain $t_0 = (0.7112, 0.0532, 0.07, 0.1656)$.

Computation with Inclusion-exclusion Formula This algorithm computes the exact resulting confidence value directly from the minimal positive and negative paths for H_0. In addition, we need the set of minimal *conflicting paths*. Therefore we set up all possible combinations consisting of one positive and one negative path ($a_x^{\pm} = a_y^{+} \cup a_z^{-}$ with $y = 1, \dots k^{+}$, $z = 1, \dots k^{-}$), minimize the set and obtain $A^{\pm} = \{a_1^{\pm}, a_2^{\pm}, \dots a_{k^{\pm}}^{\pm}\}$ (with $k^{\pm} \leq k^{+}k^{-}$). A useful optimization is to eliminate all paths that contain both H_j^{+} and H_j^{-} (because $c_j = 0$).

First, we compute the degree of conflict c_0 from the confidence values of the first-hand statements in the set of minimal paths with the inclusion-exclusion-formula ($I(A)$): c_0 is the probability that a possible world chosen according to Sect. 5.1 will contain at least one conflicting path. Thus, we add the probabilities of all minimal paths, subtract the probabilities of all unions of two minimal paths, add the probabilities of all unions of three minimal paths, etc.:

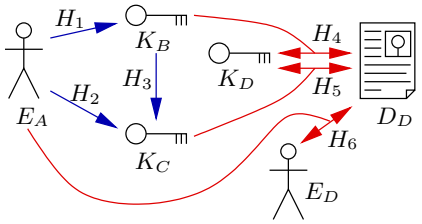
$$\begin{aligned} c_0 = & I(A^{\pm}) = \sum_{n=1}^{k^{\pm}} (-1)^{n+1} \sum_{1 \leq j_1 < \dots < j_n \leq k^{\pm}} P(a_{j_1}^{\pm} \cup \dots \cup a_{j_n}^{\pm}) \\ & = \sum_{j_1=1}^{k^{\pm}} P(a_{j_1}^{\pm}) - \sum_{1 \leq j_1 < j_2 \leq k^{\pm}} P(a_{j_1}^{\pm} \cup a_{j_2}^{\pm}) + \dots + (-1)^{k^{\pm}+1} P(a_1^{\pm} \cup \dots \cup a_{k^{\pm}}^{\pm}) \end{aligned} \tag{23}$$

$P(a)$ denotes the probability that path a is contained in the set of first-hand propositions of a chosen possible world:

$$P(a) = \prod_{j: H_j^{+} \in a \text{ or } H_j^{-} \in a} p_j \qquad \text{with } p_j = \begin{cases} 0 & \text{if } H_j^{+} \in a, H_j^{-} \in a \\ b_j & \text{if } H_j^{+} \in a, H_j^{-} \notin a \\ d_j & \text{if } H_j^{+} \notin a, H_j^{-} \in a \end{cases} \tag{24}$$

We obtain $b_0 + c_0$ with the inclusion-exclusion formula applied to the minimal positive paths, thus $b_0 = I(A^{+}) - c_0$. Similarly, the degree of disbelief is $d_0 = I(A^{-}) - c_0$ and finally we obtain $i_0 = 1 - b_0 - d_0 - c_0$.

Example 3: Authenticity Verification with Inclusion-Exclusion Formula Fig. 5 shows an example scenario consisting of the first-hand statements $H_1, \dots H_6$ with associated confidence values. Entity E_A wants to know whether entity E_D is the holder of the key K_D and therefore requests the resulting confidence value for $H_0 = \text{Auth}(E_A, K_D, E_D)$.



j	H_j	b_j	i_j	d_j
1	$\text{Trust}(E_A, K_B, r_{\text{PKI}}, 0..1)$	0.95	0.05	0
2	$\text{Trust}(E_A, K_C, r_{\text{PKI}}, 0)$	0.85	0.15	0
3	$\text{Trust}(K_B, K_C, r_{\text{PKI}}, 0)$	0.9	0.1	0
4	$\text{Auth}(K_B, K_D, D_D)$	0.7	0.2	0.1
5	$\text{Auth}(K_C, K_D, D_D)$	0.8	0.2	0
6	$\text{Auth}(E_A, D_D, E_D)$	1	0	0

Fig. 5. Scenario of Example 3

Table 6. Propositions and applied inference rules in Example 3

Proposition	Inference rule	Origin
$H^+_{1,0} = \text{Trust}^+(E_A, K_B, r_{\text{PKI}}, 0, 1)$	-	from H_1
$H^+_{1,1} = \text{Trust}^+(E_A, K_B, r_{\text{PKI}}, 1, 1)$	-	from H_1
$H^+_{2,0} = \text{Trust}^+(E_A, K_C, r_{\text{PKI}}, 0, 1)$	-	from H_2
$H^+_{3,0} = \text{Trust}^+(K_B, K_C, r_{\text{PKI}}, 0, 1)$	-	from H_3
$H^+_4 = \text{Auth}^+(K_B, K_D, D_D)$	-	from H_4
$H^-_4 = \text{Auth}^-(K_B, K_D, D_D)$	-	from H_4
$H^+_5 = \text{Auth}^+(K_C, K_D, D_D)$	-	from H_5
$H^+_6 = \text{Auth}^+(E_A, D_D, E_D)$	-	from H_6
$H^+_7 = \text{Trust}^+(E_A, K_C, r_{\text{PKI}}, 0, 2)$	(7)	$H^+_{1,1}, H^+_{3,0} \vdash H^+_7$
$H^+_8 = \text{Auth}^+(E_A, K_D, D_D)$	(20)	$H^+_{1,0}, H^+_4 \vdash H^+_8$; $H^+_{2,0}, H^+_5 \vdash H^+_8$; $H^+_7, H^+_5 \vdash H^+_8$
$H^-_8 = \text{Auth}^-(E_A, K_D, D_D)$	(20)	$H^+_{1,0}, H^-_4 \vdash H^-_8$
$H^+_0 = \text{Auth}^+(E_A, K_D, E_D)$	(17)	$H^+_6, H^+_8 \vdash H^+_0$
$H^-_0 = \text{Auth}^-(E_A, K_D, E_D)$	(17)	$H^+_6, H^-_8 \vdash H^-_0$

First, we transform the trust statements from standard form into the internal form: H_1 is transformed into $H_{1,0} = \text{Trust}(E_A, K_B, r_{\text{PKI}}, 0, 1)$ and $H_{1,1} = \text{Trust}(E_A, K_B, r_{\text{PKI}}, 1, 1)$, H_2 into $H_{2,0} = \text{Trust}(E_A, K_C, r_{\text{PKI}}, 0, 1)$, etc. Then we create the set of propositions that represents a superposition of all possible worlds according to the confidence values of the statements (see Tab. 6, $H^+_{1,0}$, $\ldots H^+_6$). Next, we apply the inference rules to the proposition of this set (including the already derived propositions). The remaining rows of Tab. 6 list the derived propositions as well as the used inference rules and the premises. The transitive trust inference rule (7) allows for example to derive the new proposition $H^+_7 = \text{Trust}^+(E_A, K_C, r_{\text{PKI}}, 0, 2)$ from $H^+_{1,1}$ and $H^+_{3,0}$. Then the minimal positive and negative paths can be determined. We find the three positive paths $\{H^+_1, H^+_4, H^+_6\}$, $\{H^+_2, H^+_5, H^+_6\}$ and $\{H^+_1, H^+_3, H^+_5, H^+_6\}$ and one negative path $\{H^+_1, H^-_4, H^+_6\}$. We can thus construct the set of minimal conflicting paths: $\{H^+_1, H^+_2, H^-_4, H^+_5, H^+_6\}$, $\{H^+_1, H^+_3, H^-_4, H^+_5, H^+_6\}$ and $\{H^+_1, H^+_4, H^-_4, H^+_6\}$. The last path can be eliminated since $c_4 = 0$.

Next we compute the degrees of conflict, belief and disbelief with the inclusion-exclusion formula: $c_0 = b_1b_2d_4b_5b_6 + b_1b_3d_4b_5b_6 - b_1b_2b_3d_4b_5b_6 = 0.07486$, $b_0 = b_1b_4b_6 + b_2b_5b_6 + b_1b_3b_5b_6 - b_1b_2b_4b_5b_6 - b_1b_3b_4b_5b_6 - b_1b_2b_3b_5b_6 + b_1b_2b_3b_4b_5b_6 - c_0 =$

$0.92358 - c_0 = 0.84872$ and $d_0 = b_1 d_4 b_6 - c_0 = 0.095 - c_0 = 0.02014$. The degree of ignorance is $i_0 = 1 - b_0 - d_0 - c_0 = 0.05628$. Thus, the resulting confidence value for H_0 is $t_0 = (0.84872, 0.05628, 0.02014, 0.07486)$.

5.3 Comparison with Other Trust Models

The model of Maurer [4] does not allow to express degrees of disbelief. Therefore, conflict can never occur. In all scenarios in which Maurer's model can be applied it produces the same resulting confidence values as our model. Subjective Logic [5] can only be applied if the network is a directed series-parallel graph (e. g., Examples 1 and 2, but not Example 3). Credential Networks [11] can be applied only if at least one component of the confidence value (b_j, i_j or d_j) of each first-hand confidence value is zero. Subjective Logic and Credential Networks produce the same results as our model in all cases in which the models and their computation approaches can be applied and in which *no conflict* occurs (e. g., in Example 1). If conflicts are possible (e. g., in Examples 2 and 3), then the results generally differ from the results of our model because these models eliminate the probability mass associated with conflict.

Our model can thus be seen as an extension of Maurer's model, Subjective Logic and Credential Networks that overcomes the mentioned restrictions ($b > 0$, $i > 0$ and $d > 0$ is possible, no restriction to directed series-parallel graphs). However, we do not eliminate the degree of conflict, because this can cause counter-intuitive effects: Consider Example 2 (Sect. 5.2). If we choose $t_1 = t_2 = t_3 = t_4 = (1, 0, 0, 0)$ (full trust), then the resulting confidence value is $t_0 = (1, 0, 0, 0)$, too (in all models). If t_4 changes to $t_4 = (0.01, 0, 0.99, 0)$ (almost complete distrust), then the resulting confidence value in our model changes to $t_0 = (0.01, 0, 0, 0.99)$, which shows E_A that the trustworthiness of E_D is now highly disputed. However, in Subjective Logic and Credential Networks the resulting confidence value does not change. This gives E_A the wrong impression that there are no trustworthy entities who distrust E_D.

6 Computation Time

Computation time is a very (although not the most) important issue for reputation systems. The computation time to find the minimal paths appears to be uncritical because it is possible to check the inference rules efficiently and because the paths can be computed incrementally and in advance. Furthermore, the paths usually remain unchanged when the confidence values of existing opinions are updated.

The number of possible worlds to consider in the possible worlds algorithm increases exponentially with the number of *relevant first-hand statements*. It is therefore applicable if the number of relevant statements is small. It is important to emphasize that the computation time depends only on the number of *relevant* statements or paths, not on the *total* number. It is sufficient to consider only statements that are issued by the requester or by authentic entities or keys that

have been found to be trustworthy. Moreover, we can ignore all statements that are not part of a valid path, i. e., that do not contribute to answer the trust or authenticity request. Furthermore, most trust chains will not be longer than two or three statements. Therefore, the number of relevant statements or paths will usually be reasonably small. Although a trust and authenticity network similar to the PGP/GnuPG web of trust [1] can contain more than 100 000 trust and authenticity statements, the number of statements that are directly or indirectly (via valid paths) related to the requester will probably be below 100, and the number of statements that are part of valid paths from the requester to the requested statement is likely to be not higher than 10 or 20 in typical scenarios.

The number of possible worlds in the grouped possible worlds algorithm increases exponentially with the number of *groups*. Thus, the computation time can be reduced significantly if the statements can be grouped. Even large scenarios can be evaluated efficiently as long as the relevant statements can be subdivided into a small number of groups. In the inclusion-exclusion algorithm the number of summands increases exponentially with the number of relevant *paths*. This algorithm is therefore well suited for all scenarios with a small number of paths, even if the number of statements is large.

We illustrate the influence of the scenario (i. e., the number of relevant statements, paths and groups) on the computation time of the algorithms on two examples[5]. The scenarios are constructed in order to emphasize the large influence on the computation time and are not meant to be representative examples. For simplicity the scenarios consist only of trust statements between entities and refer to the same capability r. All confidence values contain degrees of belief, ignorance and disbelief ($b > 0, i > 0, d > 0$).

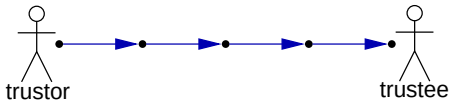


Fig. 6. Scenario of the simple chain example

The scenario with e entities in Fig. 6 consists of a simple chain and $e - 1$ trust statements with $h = 0..e$ recommendation hops. The possible worlds algorithm has to evaluate 3^{e-1} worlds. The scenario contains one positive and one negative path, therefore the inclusion-exclusion algorithm has to compute only two summands. The grouped possible world algorithm creates one group with three possible proposition-combinations: the positive path leads to belief, the negative path to disbelief, all other combinations lead to ignorance. It thus has to evaluate only three worlds. The diagram in Fig. 7 shows that the computation time of the possible world algorithm increases exponentially with the number of trust statements, whereas the computation time of the other algorithms increases linearly and thus remains insignificant even for long trust chains.

[5] implementation in Java 1.6; measurements on Intel Pentium M with 1.73 GHz

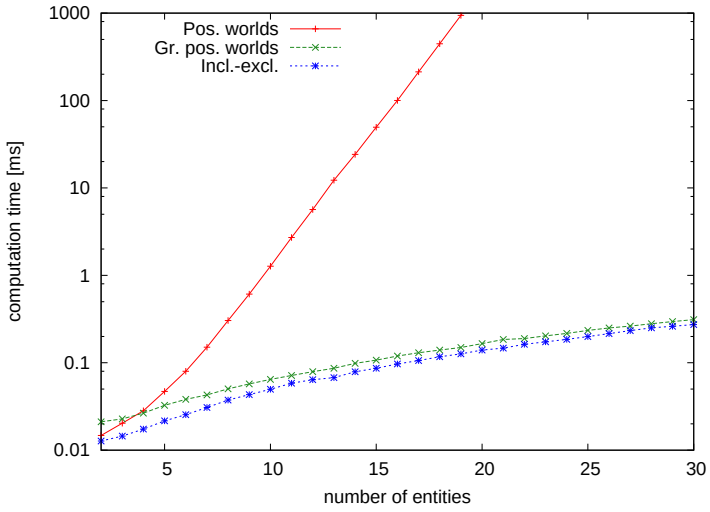


Fig. 7. Computation time in the simple chain example

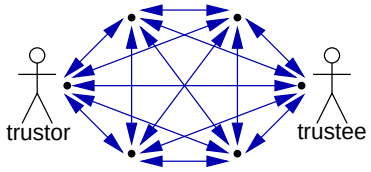


Fig. 8. Scenario of the full mesh example

The scenario in Fig. 8 is a full mesh. All entities trust each other for $h = 0..1$ hops. The number of relevant statements is $2e - 3$, the number of positive and negative paths is $e - 1$ and the number of conflicting paths is $(e - 1)(e - 2)$. Thus, the computation time of the possible worlds algorithm increases slower than of the inclusion-exclusion algorithm because the number of conflicting paths increases faster than the number of relevant statements (Fig. 9). The grouped possible worlds algorithm subdivides the statements into $e - 1$ groups, which reduces the number of possible worlds from 3^{2e-3} to 3^{e-1} worlds. Therefore the computation time remains acceptable for a larger number of entities than with the other algorithms.

The computation time heavily depends on the scenario. It is therefore difficult to give a general recommendation for one of the algorithms. It is possible that one algorithm outperforms an other by orders of magnitude in one scenario, but is much slower in an other scenario. The presented results and measurements in other scenarios suggest that the grouped possible worlds algorithm is in most scenarios the fastest (or at least close to the fastest) algorithm. However, further investigations are necessary.

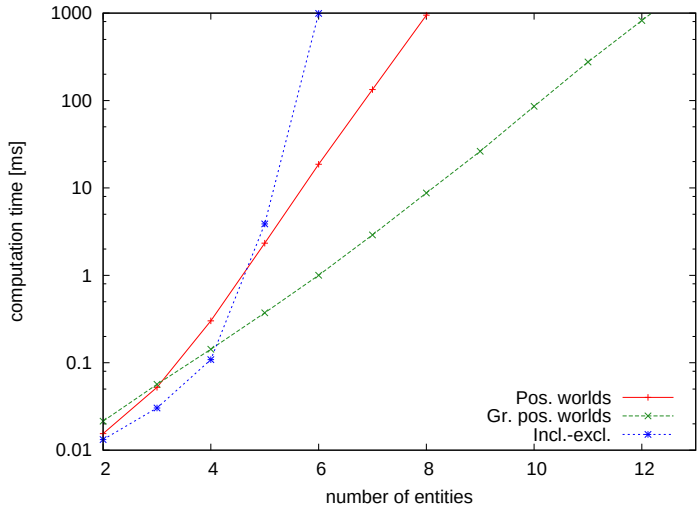


Fig. 9. Computation time in the full mesh example

An estimation for the computation time of the algorithms can be computed from the number of relevant statements, paths and groups. If the expected computation of all exact algorithms exceeds an acceptable limit, then a Monte-Carlo simulation can be used to compute an approximation. The number of iterations can be chosen according to the required accuracy and the acceptable computation time.

7 Summary and Conclusions

We presented a detailed model to represent trust and authenticity statements as well as confidence values and we proposed an integrated approach to reason with these statements and to compute resulting confidence values. The model distinguishes clearly between entities, descriptions and keys, allows multiple keys and descriptions per entity, distinguishes between functional and recommendation trust and allows to specify ranges of recommendation hops in trust statements. Confidence values allow to express degrees of belief, ignorance and disbelief. The system is able to reason with conflicting opinions because the presented inference rules are based on a paraconsistent logic. The computation of the resulting confidence values is based on a probability theoretical model in order to produce consistent results. In conflict-free scenarios our model is consistent with the Model of Maurer, Subjective Logic and Credential Networks, but overcomes several restrictions of these models. In conflicting scenarios we do not eliminate the degree of conflict in order to avoid counter-intuitive effects caused by re-normalizations.

We proposed different algorithms to implement the confidence value computation. Although the computation time increases exponentially with the number

of relevant statements, groups or paths it can be expected that an acceptable computation time can be reached in the majority of realistic scenarios. In the other cases, we propose to compute an approximation with Monte-Carlo simulations.

Acknowledgements This work was funded by the German Research Foundation (DFG) through the Collaborative Research Center (SFB) 627.

References

1. Ashley, J.M., Copeland, M., Grahn, J., Wheeler, D.A.: The GNU Privacy Handbook. The Free Software Foundation. (1999)
2. Grandison, T., Sloman, M.: A Survey of Trust in Internet Application. IEEE Communications Surveys & Tutorials **3**(4) (2000) 2–16
3. Marsh, S.P.: Formalising Trust as a Computational Concept. PhD thesis, Department of Mathematics and Computer Science, University of Stirling (1994)
4. Maurer, U.: Modelling a Public-Key Infrastructure. In: Proc. 1996 European Symposium on Research in Computer Security (ESORICS' 96). Volume 1146 of Lecture Notes in Computer Science., Springer-Verlag (1996) 325–350
5. Jøsang, A.: Artificial Reasoning with Subjective Logic. In: Proceedings of the Second Australian Workshop on Commonsense Reasoning. (1997)
6. Haenni, R., Kohlas, J., Lehmann, N. In: Probabilistic Argumentation Systems. Volume 5 (Algorithms for Uncertainty and Defeasible Reasoning) of Handbook of Defeasible Reasoning and Uncertainty Management Systems. Springer (2000) 221–288
7. Kamvar, S.D., Schlosser, M.T., Garcia-Molina, H.: The EigenTrust Algorithm for Reputation Management in P2P Networks. In: Proceedings of the 12th International Conference on World Wide Web. (2003) 640–651
8. Demolombe, R.: Reasoning about trust: A formal logical framework. In: Proceedings of the Second International Conference of Trust Management (iTrust 2004). (2004) 291–303
9. Jøsang, A., Ismail, R., Boyd, C.: A survey of trust and reputation systems for online service provision. In: Decision Support Systems. (2007)
10. Kohlas, R.: Decentralized Trust Evaluation and Public-Key Authentication. PhD thesis, Universität Bern (2007)
11. Jonczy, J., Haenni, R.: Credential Networks: a General Model for Distributed Trust and Authenticity Management. In: PST. (2005) 101–112
12. Jøsang, A., Gray, E., Kinateder, M.: Simplification and Analysis of Transitive Trust Networks. Web Intelligence and Agent Systems Journal (2006) 139–161
13. Zadeh, L.A.: Review of Books: A Mathematical Theory of Evidence. The AI Magazine **5**(3) (1984) 81–83
14. Shafer, G.: A Mathematical Theory of Evidence. Princeton Univ. Press (1976)
15. Gutscher, A.: A Trust Model for an Open, Decentralized Reputation System. In: Proceedings of the Joint iTrust and PST Conferences on Privacy Trust Management and Security (IFIPTM 2007). (2007) 285–300
16. Gutscher, A.: Reasoning with Uncertain and Conflicting Opinions in Open Reputation Systems. In: Proceedings of the Fourth International Workshop on Security and Trust Management (STM 2008). (2008)