Semantic Constraints for Trust Transitivity

Audun Jøsang **Simon Pope**

Distributed Systems Technology Centre*
University of Queensland, Brisbane, Australia
Email: {ajosang, simon.pope}@dstc.edu.au

Abstract

To describe the concept of transitive trust in a simplified way, assume that agent A trusts agent B, and that agent B trusts agent C, then by transitivity, agent A trusts agent C. Trust transitivity manifests itself in various forms during real life human interaction, but can be challenging to concisely model in a formal way. In this paper we describe principles for expressing and analysing transitive trust networks, and define requirements for their validity. This framework can be used for modelling transitive trust in computerised interactions, and can be combined with algebras and algorithms for computing propagation of both trust and distrust. This is illustrated by an example where transitive trust is mathematically analysed with belief calculus.

1 Introduction

Computer networks are increasingly removing us from a familiar direct style of interacting. We may now collaborate online with people or organisations we have never met, and perhaps never heard of before. Many of the traditional strategies for representing and assessing trustworthiness can no longer be used in those situations. It can therefore be difficult to assess whether the services and information provided by a remote party are reliable or whether they are correctly represented by the remote party. Organisations involved in online service provision also face the challenge of building online systems and online customer relationships which engender trust.

There is thus a need for mechanisms that enable relying parties to determine the trustworthiness of remote parties through computer mediated communication and collaboration. These mechanisms should also allow trustworthy entities to be recognised as such. The idea is that such trust and reputation systems will enable highly trustworthy entities to attract collaboration from others, and discourage low quality and fraudulent players from participating in the community, or alternatively encourage them to behave in a more trustworthy manner.

Agents are well suited to take advantage of automated trust systems. This will allow more reliable agent-to-agent and agent-to-human interact. In particular trust systems are important factor for creating stable agent communities where deceitful behaviour is possible[15]. Being able to formally express and reason with trust is necessary for allowing humans and agents assessing trust in electronic environments. The aim of this will be to create communication infrastructures where trust can thrive in order to ensure meaningful and mutually beneficial interactions between players.

In this regard, we intend to describe semantic criteria and a notation for specifying networks of transitive trust. We first consider properties of trust diversity, transitivity, and parallel combination. We then define a notation for describing and reasoning about trust, and illustrate how this notation may successfully and securely be used to correctly analyse trust networks. We identify several requirements that trust measures and operators should satisfy, and finally provide an example of how belief calculus can be used as a practical method for computing transitive trust.

2 Defining Trust

Manifestations of trust are easy to recognise because we experience and rely on it every day, but at the same time trust is quite challenging to define because it manifests itself in many different forms. The literature on trust can also be quite confusing because the term is being used with a variety of meanings [22]. Two different types of trust which we will call *reliability trust* and *decision trust* respectively are commonly encountered in the literature.

As the name suggest, reliability trust can be interpreted as the reliability of something or somebody, and the definition by Gambetta (1988) [7] provides an example of how this can be formulated:

Definition 1 (Reliability Trust) *Trust is the subjective probability by which an individual, A, expects that another individual, B, performs a given action on which its welfare depends.*

This definition includes the concept of *dependence* on the trusted party, and the *reliability* (probability) of the trusted party as seen by the trusting party.

However, trust can be more complex than Gambetta's definition indicates. For example, Falcone & Castelfranchi (2001) [6] recognise that having high (reliability) trust in a person in general is not necessarily enough to decide to enter into a situation of dependence on that person. In [6] they write: *"For example it is possible that the value of the damage per se (in case of failure) is too high to choose a given decision branch, and this independently either from the probability of the failure (even if it is very low) or from the possible payoff (even if it is very high). In other words, that danger might seem to the agent an intolerable risk."* In order to capture this broad concept of trust, the following definition from McKnight & Chervany (1996) [22] can be used.

Definition 2 (Decision Trust) *Trust is the extent to which one party is willing to depend on the other party in a given situation with a feeling of relative security, even though negative consequences are possible.*

*The work reported in this paper has been funded in part by the Co-operative Research Centre for Enterprise Distributed Systems Technology (DSTC) through the Australian Federal Government's CRC Programme (Department of Education, Science, and Training).

The relative vagueness of this definition is useful because it makes it the more general. It explicitly and implicitly includes aspects of a broad notion of trust which are *dependence* on the trusted entity or party, the *reliability* of the trusted entity or party, *utility* in the sense that positive utility will result from a positive outcome, and negative utility will result from a negative outcome, and finally a certain *risk attitude* in the sense that the trusting party is willing to accept the situational risk resulting from the previous elements. Risk emerges for example when the value at stake in a transaction is high and the probability of failure is non-negligible. Contextual aspects such law enforcement, insurance and other remedies in case something goes wrong are only implicitly included in the definition of trust above, but should nevertheless be considered to be part of trust.

We consider reliability trust to be the most appropriate for the purpose of describing trust transitivity. This also means that actual transaction utilities and risk attitudes will not be explicitly included in the formalism we will present here. However, these aspects can be considered implicitly included in the *trust purpose*.

We will use the term *"trust purpose"* to express the semantic content of an instantiation of trust. A particular trust purpose can for example be *"to be a good car mechanic"* in the sense that a party A can trust a party B for that purpose. When the trust purpose relates to an item, it describes the belief that a material or abstract object has a given property or is in a given state. This allows trust to be applied to non-living things such as for example trust in the safety of a car or trust in the integrity of a data file. When the trust purpose relates to a living (human) agent, it describes the belief that the agent will behave in one's best interest, i.e. that it can be depended upon. An agent can be anything from an individual person to a human organisation consisting of thousands of individuals.

In order for trust to form transitive networks, it needs to be expressed with three basic diversity dimensions [9] where the first dimension represents the trustor or trust originator, the second represents the trust purpose, and the third represents the trustee or the trust target. This is illustrated in Fig.1 below.

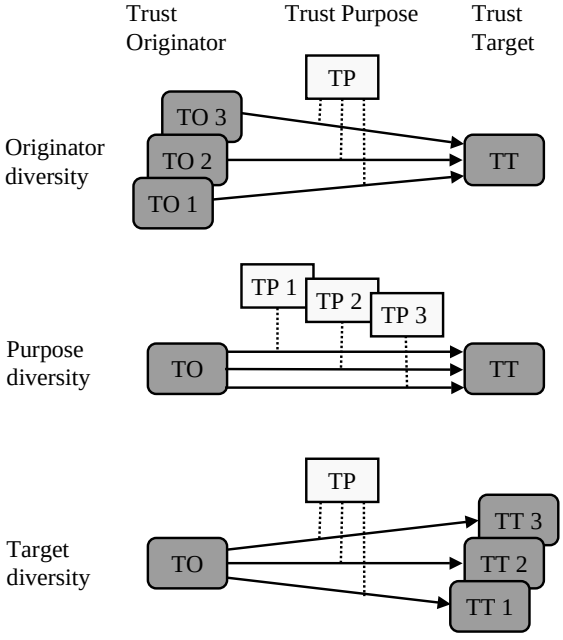

Figure 1: Basic trust diversity

In addition to the three basic trust dimensions, a *trust measure* can be associated with each trust relationship. The trust measure could for example be binary (trusted, not trusted), discrete (e.g. strong trust, weak trust, strong distrust, weak distrust, etc.) or continuous in some form (e.g. probability, percentage or belief functions of trustworthiness). The topic of expressing and computing trust measures will be discussed in Sec. 8.

In addition, a fifth important element to a trust relationship is its *time* component. Quite obviously trust of the trustor in the trustee regarding a certain purpose at one point in time might be quite different from this trust after several transactions between these two entities have taken place. This means, that we can model time as a set of discrete events taking place, where both the trustor and the trustee are involved. However, even if no transactions take place, a trust relationship will gradually change with time passing. Therefore, in addition to the discrete changes that are made when events have occurred, we must also take into account continuous changes to trust relationships.

3 Trust Transitivity

It has been shown [5] that trust is not always transitive in real life. For example the fact that Alice trusts Bob to fix her car, and Bob trusts Claire to look after his child, does not imply that Alice trusts Claire for fixing the car, or for looking after her child. However, under certain semantic constraints, trust can be transitive, and a trust referral system can be used to derive transitive trust. In the above example, the trust transitivity broke down because Alice and Bob's trust purposes were different and did not fit together.

Let us now assume that Alice needs to have her car serviced, and that she asks Bob for his advice about where to find a good car mechanic in town. Bob is thus trusted by Alice to know about a good car mechanic and to tell his honest opinion about that, whereas Bob actually trusts the car mechanic. Through trust transitivity, Alice will also trust that car mechanic when Bob recommends him to her.

Let us make the example slightly more complicated, wherein Bob does not actually know any car mechanics himself, but he knows Claire whom he believes knows a good car mechanic. As it happens, Claire is happy to recommend the car mechanic named David, and as a result of trust transitivity through Bob and Claire, Alice also trusts David, as illustrated in Fig.2 below.

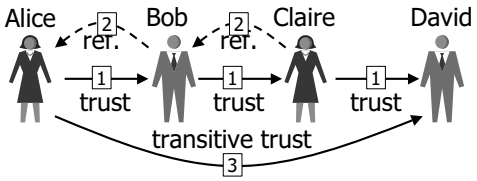

Figure 2: Transitive trust derivation

The indexed arrows in Fig.2 indicate the order in which trust links and referrals are formed. The initial trust links have index 1. By assuming Alice's trust in Bob and Bob's trust in Claire to be positive, Alice's derived trust in David is intuitively also positive.

Claire obviously recommends to Bob her opinion about David as a car mechanic, but Bob's referral to Alice is ambiguous. Bob could pass Claire's referral unaltered on to Alice, meaning that he repeats exactly what Claire said. Bob could also derive indirect functional trust in David, and give a recommendation to Alice based on that trust. The latter way of passing referrals can create problems and it is better for Alice when she receives Claire's referral unaltered. This will be discussed in more detail in Sec. 6.

The trust originators and targets in Fig.2 are explicit, but it is challenging to define exactly for what purpose Alice for example trusts Bob. The most obvious is to say that Alice trusts Bob to recommend somebody who can recommend a good car mechanic. The problem with this

type of formulation is that the length of the trust purpose expression becomes proportional with the length of the transitive path, so that the trust purpose rapidly becomes an impenetrable expression. It can be observed that this type of trust purpose has a recursive structure that can be exploited to define a more compact expression for the trust purpose. Trust in the ability to refer to a third party, which can be called *referral trust*, is precisely what allows trust to become transitive. At the same time this trust always assumes the existence of *functional trust* at the end of the transitive path, which in the example above is about being a good car mechanic.

Alice would then have referral trust in Bob to be a good car mechanic, and the same for Bob's trust in Claire. This must be interpreted as saying that Alice trusts Bob to recommend somebody (to recommend somebody etc.) to be a good car mechanic. Obviously it does not mean that Alice trusts Bob to actually be a good car mechanic. On the other hand Claire would have *functional* trust in David to be a good car mechanic, which means that she actually believes he is a good car mechanic. The "referral" variant of a trust purpose is recursive so that any transitive trust path, with arbitrary length, can be expressed using only two variants of a single trust purpose.

The idea of constructing paths of transitive trust based on a single trust purpose with functional and referral variants is captured by the following definition.

Definition 3 (Matching Trust Purposes) *A valid transitive trust path requires that the last edge in the path represents functional trust and that all other edges in the path represent referral trust, where the functional and the referral trust edges all have the same trust purpose.*

A transitive trust path therefore stops with the first functional trust edge encountered and when there are no more outgoing referral trust edges. It is of course possible for a principal to have both functional and referral trust in another principal but that should be expressed as two separate trust edges. The existence of both a functional and an referral trust edge e.g. from Claire to David should be interpreted as Claire having trust in David not only to be a good car mechanic but also to recommend other car mechanics.

The examples above assume some sort of absolute trust between the agents in the transitive path. In reality trust is never absolute, and many researchers have proposed to express trust as discrete verbal statements, as probabilities or other continuous measures. One observation which can be made from a human perspective is that trust is weakened or diluted through transitivity. By taking the example above, it means that Alice's derived trust in the car mechanic David through the recommenders Bob and Claire can be at most as strong as Claire's trust in David.

It could be argued that negative trust within a transitive path can have the paradoxical effect of strengthening the derived trust. Take for example the case where Alice trusts Bob, Bob distrusts Claire, and Claire distrusts David. In this situation Alice might actually derive positive trust in David, since she relies on Bob's advice and Bob says: "Claire is a cheater, do not rely on her!". So the fact that Claire distrusts David might count as a pro-David argument from Alice's perspective. The question boils down to "is the enemy of my enemy my friend?". However this question relates to how trust is computed and derived, and this will be discussed in Sec.8

4 Parallel Trust Combination

It is common to collect referrals from several sources in order to be better informed when making decisions. This can be modelled as *parallel trust combination*.

Let us assume again that Alice needs to get her car serviced, and that she asks Bob to recommend a good car mechanic. When Bob recommends David, Alice would like to get a second opinion, so she asks Claire for her opinion about David. Intuitively, if both Bob and Claire recommend David to be a good car mechanic, Alice's trust in David will be stronger than if she had only asked Bob. Parallel combination of positive trust thus has the effect of strengthening the derived trust. This is illustrated in Fig.3 below.

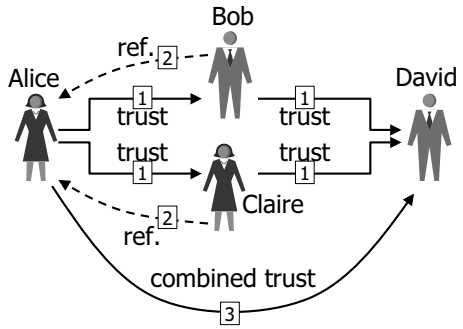

Figure 3: Combination of parallel trust paths

In the case where Alice receives conflicting recommended trust, e.g. both trust and distrust, then she needs some method for combining these conflicting referrals in order to derive her trust in David.

5 Structured Notation

Transitive trust networks can involve many principals, and in the examples below, capital letters A, B, C, D, E and F will be used to denote principals instead of names such as Alice and Bob.

We will use basic constructs of directed graphs to represent transitive trust networks. We will add some notation elements which allow us to express trust networks in a structured way.

A single trust relationship can be expressed as an edge, where the vertices are the trust originator and the trust target respectively. For example the edge (A, B) means that A trusts B.

The symbol ":" will be used to denote the transitive connection of two consecutive trust edges to form a transitive trust path.

The trust relationships of Fig.2 can be expressed as:

$$(A, D) = (A, B) : (B, C) : (C, D) \tag{1}$$

where the trust purpose is implicit. Let the trust purpose be defined as P; *"trusts X to be a good car mechanic"*. Let the functional variant be denoted by fP and the referral variant by rP. A distinction can be made between initial *direct trust* and derived *indirect trust*. Whenever relevant, this can be indicated by prefixing the letter d to the trust purpose to indicate direct trust (dP), and to prefix the letter i to the trust purpose to indicate indirect trust (iP). This can be combined with referral and functional trust so that for example indirect functional trust can be denoted as ifP. The trust purpose can then be explicitly included in the trust edge notation as e.g. denoted by (A, B, drP).

The trust network of Fig.2 can then be explicitly expressed as:

$$\begin{aligned}&(A, D, ifP)\\ &= (A, B, drP) : (B, C, drP) : (C, D, dfP)\end{aligned} \tag{2}$$

Let us now turn to the combination of parallel trust paths, as illustrated in Fig.3. We will use the symbol "$\diamond$" to denote the connector for this purpose. The "$\diamond$" symbol

visually resembles a simple graph of two parallel paths, so that it is natural to use it in this context.

Alice's combination of the two parallel trust paths from her to David in Fig.3 can then be expressed as:

$$(A, D, ifP) = \begin{array}{r} ((A, B, drP) : (B, D, dfP)) \\ \diamond ((A, C, drP) : (C, D, dfP)) \end{array} \quad (3)$$

Let trust measures be denoted by μ_i where i refers to a specific trust measure, and let Alice, Bob and Claire's trust measures be μ_1, μ_2 and μ_3 respectively. Let time stamps be denoted by τ_j where j refers to a specific time, and let the trust measures be time stamped τ_1, τ_2 and τ_2 respectively. Alice's derived trust measure and time stamp are denoted by μ_4 and τ_4. The derived trust from the trust path of Fig.2 can then be expressed as:

$$(A, D, ifP, \mu_4, \tau_4) = \begin{array}{r} (A, B, drP, \mu_1, \tau_1) \\ : (B, C, drP, \mu_2, \tau_2) \\ : (C, D, dfP, \mu_3, \tau_3) \end{array} \quad (4)$$

With this structured notation, arbitrarily large trust networks can be expressed. In most computational frameworks it will be required that any trust edge can only appear once in the expression of a trust graph, because otherwise, dependence and incorrectly derived trust measures could be the result, as explained below.

6 Trust Network Analysis

We will first explain why a referral should always be passed in its original form from the recommender to the relying party, and not as indirect derived trust. Fig.4 shows an example of how not to provide referrals.

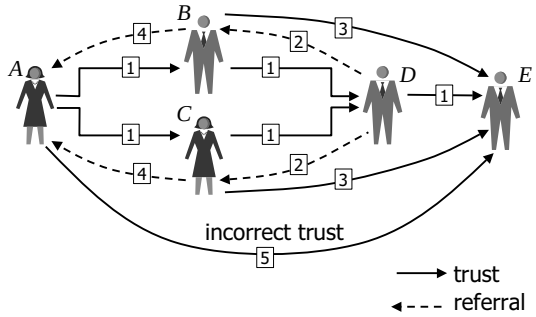

Figure 4: Incorrect analysis

In Fig.4 the trust and referral arrows are indexed according to the order in which they are formed. In the scenario of Fig.4, the initial trust links have index 1. First D passes his referral about E to B and C (index 2) so that B and C are able to derive indirect trust in E (index 3). Now B and C pass their derived indirect trust in E to A (index 4) so that she can derive indirect trust in E (index 5). As a result, A perceives the trust network between her and E to be

$$(A, E) = ((A, B) : (B, E)) \diamond ((A, C) : (C, E)) \quad (5)$$

The problem with this scenario is that A is ignorant about the fact that $(B, E) = (B, D) : (D, E)$ and that $(C, E) = (C, D) : (D, E)$ so that A in fact derives the following hidden trust network:

$$(A, E) = \begin{array}{r} ((A, B) : (B, D) : (D, E)) \\ \diamond ((A, C) : (C, D) : (D, E)) \end{array} \quad (6)$$

It can be seen that the hidden trust network contains dependencies because the edge (D, E) appears twice. Neither the perceived nor the hidden trust network is equal to the real trust network, which indicates that this way of passing referrals can produce incorrect results.

We argue that B and C should pass the referrals explicitly as $(B, D) : (D, E)$ and $(C, D) : (D, E)$ respectively, and this is certainly possible, but then A needs to be convinced that B and C have not altered the referral (D, E) that they received from D. If B and C are dishonest, they might for example try to change the recommended trust measures related to the trust edge (D, E). Not only that, any party that is able to intercept the referrals sent to A might want to alter the trust values, and A needs to receive evidence of the authenticity and integrity of the referrals. Cryptographic security mechanisms can typically be used to solve this problem, and this will be discussed in more detail in Sec.7.

It is thus necessary that A receives direct trust referrals unaltered and as expressed by the original recommending party. An example of a correct way of passing referrals is indicated in Fig.5

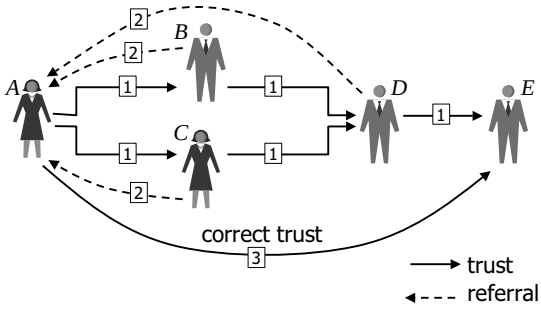

Figure 5: Correct analysis

In the scenario of Fig.5 the trust network perceived by A is equal to the real trust network which can be expressed as:

$$(A, E) = \begin{array}{r} (((A, B) : (B, D)) \\ \diamond ((A, C) : (C, D))) : (D, E) \end{array} \quad (7)$$

The lesson to be learned from the scenarios in Fig.4 and Fig.5 is that there is a crucial difference between recommending trust in a principal resulting from your own experience with that principal and recommending trust in a principal which has been derived as a result of referrals from others. We will use the term *direct trust* to denote the former, and *indirect trust* for the latter. Fig.4 illustrated how problems can occur when indirect trust is recommended, so the rule is to only recommend direct trust [10]. For example, A's derived indirect trust in E in Fig.5 should not be recommended to others.

Expressing transitive trust graphs in the form of e.g. Eq. (7) is not always practical, for example when the trust network is complex, or when only parts of it are known. Instead, each isolated trust relationship can be expressed individually, and an automated parser can establish valid trust paths and graphs depending on the need.

The initial direct trust relationships of Fig.5 can for example be listed as in Table 1 below.

A parser going through Table 1 will be able to determine the trust network of Fig.5. The principal A can be called a relying party because she relies on the referrals from B, C and D to derive her trust in E. We will assume that relying parties will always try to base derived trust on the most recent referrals. In Table 1 it can be observed that there are two entries for the trust edge (A, B), and based on the principle of the most recent trust expression, the parser would select the last entry expressed by

Table 1: Initial direct trust relationships of Fig.5

Edge	Purp.	Vari.	Meas.	Time
(A, B)	P	r	μ_1	$\tau_1 = 31.01.2005$
(A, C)	P	r	μ_2	$\tau_1 = 31.01.2005$
(B, D)	P	r	μ_3	$\tau_1 = 31.01.2005$
(C, D)	P	r	μ_4	$\tau_1 = 31.01.2005$
(D, E)	P	f	μ_5	$\tau_1 = 31.01.2005$
(A, B)	P	r	μ_6	$\tau_2 = 01.02.2005$

$(A, B, rP, \mu_6, \tau_2)$. If the relying party A derives her trust in E at or after τ_2, then that trust can be expressed as:

$$
\begin{aligned}
&(A, E, ifP_1, \mu_7, \tau_2) = \\
&\quad (((A, B, drP, \mu_6, \tau_2) : (B, D, drP, \mu_3, \tau_1)) \\
&\quad \diamond ((A, C, drP, \mu_2, \tau_1) : (C, D, drP, \mu_4, \tau_1))) \\
&\quad : (D, E, dfP, \mu_5, \tau_1)
\end{aligned} \tag{8}
$$

In this example, the time stamp of the derived trust is set equal to the most recent time stamp of all the trust measures used in the computation in order to give an indication of the freshness of the input data.

7 Integrity and Authenticity of Trust Referrals

Cryptography can be used to provide authenticity and integrity of trust referrals. This in turn requires that every participant holds a trusted (i.e. authentic) key. The process of generating, distributing and using cryptographic keys is called key management, and this still is a major and largely unsolved problem on the Internet today.

Public-key infrastructures (PKI) simplify key management and distribution, but has very strict trust requirements. A PKI refers to an infrastructure for distributing public keys where the authenticity of public keys is certified by Certification Authorities (CA). A certificate basically consists of the CA's digital signature on the public key together with the owner identity, thereby linking the key and the owner identity together in an unambiguous way. In order to verify a certificate, the CA's public key is needed, thereby creating an identical authentication problem. The CA's public key can be certified by another CA etc., but in the end you need to receive the public key of some CA out-of-band in a secure way. Although out-of-band channels can be expensive to set up and operate they are absolutely essential in order to obtain a complete path of trust from the relying party to the target public key.

However, there are potential trust problems in this design. What happens if a CA issues a certificate but does not properly check the identity of the owner, or worse, what happens if a CA deliberately issues a certificate to someone with a false owner identity? Furthermore, what happens if a private key with a corresponding public-key certificate is leaked to the public domain by accident, or worse, by intent? Such events could lead to systems and users making totally wrong assumptions about identities in computer networks. Clearly CAs must be trusted to be honest and to do their job properly and users must be trusted to protect their private keys.

When including security in the description of our scheme, it must be assumed that every principal has a public/private key pair that can be used for authentication and encryption. We can either assume that the public keys are absolutely trusted (i.e. that the relying party is absolutely certain about their authenticity) or that they too can have various levels of trustworthiness. The easiest is of course to assume absolute trust, because then the authenticity and integrity of the trust referrals can be assumed, and trust networks can be analysed as described in the previous sections.

If on the other hand trust in cryptographic keys can have varying measures, then the trust in every cryptographic key must be determined before the primary trust network of interest can be analysed. Trust in public keys can be derived from trust in the various components of a PKI. A method for analysing trust in the authenticity of public keys in a PKI is described in detail in [10]. We build on that model by explicitly separating between the functional and referral trust purposes, as illustrated in Fig.6 below.

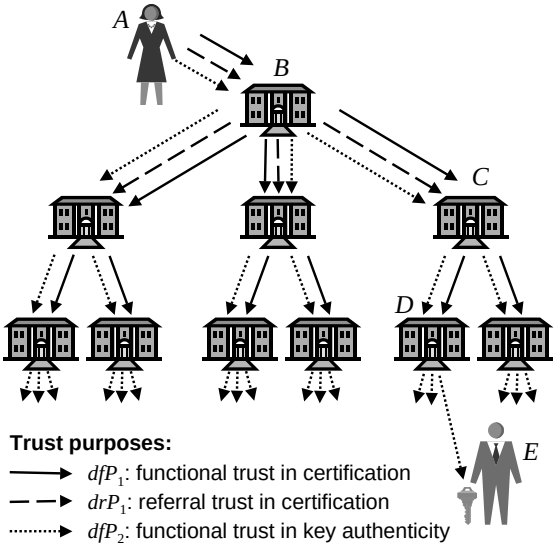

Figure 6: Structure of trust purposes in PKI

Two different trust purposes are used:

- P_1: *"Trusts the CA to correctly certify public keys"*
- P_2: *"Trusts the public key to be authentic"*.

In Fig.6 it is assumed that the relying party A has obtained the public key of B. Since B is a root CA, Alice must have obtained the key through a secure out-of-band channel. Depending on the security of that out-of-band channel, Alice has a level of direct functional trust in the authenticity of B's public key, which can be denoted by (A, B, dfP_2). In addition, A has direct functional trust in B to correctly certify the public keys of subordinate CAs, which can be denoted by (A, B, dfP_1). Finally A has direct referral trust in B for the same purpose, meaning that A trusts B to verify that subordinate CAs are able to correctly certify public keys, which can be denoted by (A, B, drP_1).

B needs to trust C for exactly the same purposes as A trusts B. The trust that C has in D does not need the referral variant of P_1 because there are no subordinate CAs under D.

D's trust in the user E is the simplest, because it only focuses on the authenticity of E's public-key, which can be denoted by (D, E, dfP_2).

The relying party A is interested in deriving a measure of authenticity of user E's public key through the trust web of this PKI. With the specified trust purposes and trust relationships, this can be expressed as:

$$
\begin{aligned}
&(A, E, ifP_2) = \\
&\quad ((A, B, dfP_2) \wedge (A, B, dfP_1) \wedge (A, B, drP_1)) \\
&\quad : ((B, C, dfP_2) \wedge (B, C, dfP_1) \wedge (B, C, drP_1)) \\
&\quad : ((C, D, dfP_2) \wedge (C, D, dfP_1)) \\
&\quad : (D, E, dfP_2)
\end{aligned} \tag{9}
$$

The existence of up to three separate trust edges between parties requires some method for combining them together. Various methods can be imagined for this purpose and one possibility is to use conjunction (i.e. logical AND in the binary case) of the two trust purposes[10]. The connector "$\wedge$" is used in Eq.(9) to denote that a conjunction of the trust purposes is needed, e.g. meaning that A must trust B to have an authentic key, AND to provide reliable certification, AND to verify that subordinate CAs also provide reliable certification.

The consequence of having to derive trust in public keys is that the relying party might have to analyse a separate auxiliary trust network for every principal in the trust network of interest. Deriving indirect trust in a remote party would then have to take the authenticity of the public keys into account in addition to the trust in the principal. We will illustrate this with a very simple example, such as for delivering an online service, where A received a trust referral from E about F for a particular purpose. The trust relationships that have to be taken into account are illustrated in Fig.7 below. The parties A and E are the same as those in Fig.6, and A is now using the trust she derived from the PKI. Two different trust purposes are used:

- P_2: *"Trusts the public key to be authentic"*.
- P_3: *"Trusts the agent to provide quality service"*

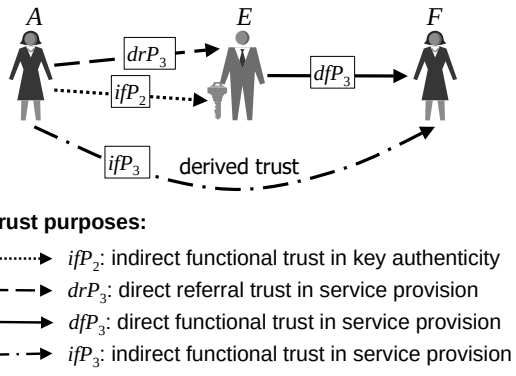

Figure 7: Trust transitivity with authenticated public keys

We will again use the symbol "$\wedge$" to denote conjunction of two required trust purposes between the same pair of entities. A's derived trust in service provider F can then be expressed as:

$$(A, F, ifP_3) = \\ ((A, E, ifP_2) \wedge (A, E, drP_3)) : (E, F, dfP_3) \quad (10)$$

For a parser to be able to derive this trust network, it is of course required that the relying party A has received and stored all these trust referrals for example in the form of a table similar to Table 1. Only the first trust purpose in a conjunctive trust relationship is used by the parser to determine the actual trust network. The second trust purpose is only used when computing the derived trust measure.

To illustrate the role of key authenticity, take for example the case when a principal is recommended to be reliable but that the binding between the principal and his key is broken, e.g. because it is known that the private key has been stolen by an intruder. The conjunction between trust in the principal and the distrust in his key would result in reduced trust, indicating that a principal identified by this particular public key can not be strongly trusted despite an otherwise positive trust referral. This is what intuition would dictate because it is now possible that referrals that appear to come from the principal in fact originate from the intruder who stole the private key and who is not trusted.

8 Measuring and Computing Trust

In previous sections we have indicated some intuitive principles that trust measures and computational rules should follow. This section describes additional requirements that trust measures and operators should satisfy. Sec.9 below describes a practical example of how measures of trust can be mathematically expressed and derived.

While trust has no natural or physical measurable units, its value can be measured as subjective probability in the simplest models, or as a function of multiple components such as reliability, utilities and risk attitudes in more complex models [16, 23]. Many trust measures and trust derivation schemes have been proposed in the literature varying from discrete measures [25, 1, 3, 4, 20] to continuous measures [21, 2, 10, 11, 19, 8, 18, 24].

Typical discrete trust measures are for example "strong trust", "weak trust", strong distrust" and "weak distrust". PGP[25] is a well known software tool for cryptographic key management and email security that for example uses the discrete trust measures "ultimate", "always trusted", "usually trusted", "usually not trusted" and "undefined" for key owner trust. In order to obtain compatibility between discrete and continuous methods it should be possible to interpret such discrete verbal statements by mapping them to continuous measures.

When measuring trust, it is critical that the trust value is *meaningful* to and *usable* by both the originator and the target transacting partners. Otherwise, if trust is subjectively measured by each party using different methods and scales, the value becomes meaningless and unusable. By explicitly defining P_1 and P_2 in the scenarios above, we ensure that the interacting parties have a common understanding of the trust purpose, which is a prerequisite for deriving meaningful trust values for one another.

As mentioned in Sec. 2, *time* is an element that should be captured together with trust measures. This element is necessary not only to demonstrate how trust is evolving, but also in order to enable transaction partners to assess trust based on, for example, the most recent trust value available.

Determining the *confidence* of the trust measure is also a requirement. For example, the weakening of trust through long transitive paths should result in a reduced confidence level, and not necessarily lead to distrust. On the other hand, a large number of parallel referrals should result in an increased confidence level.

Finally, in order to derive trust measures from a trust network there must be explicit methods for combining the trust measures along a transitive path as in Fig.2, for combining the trust measures of parallel paths as in Fig.3 as well as for combining trust measures in a conjunction of trust relationships as in Fig.7. Various methods and principles for deriving trust from such combinations have been proposed in the literature [10, 25, 1, 4, 11, 19, 18]. The validation and suitability assessment of any computational approach should be based on simulations and usability studies in environments equal or similar to those where it is intended for deployment.

9 Trust Derivation with Subjective Logic

Belief calculus is a mathematical framework that provides operators that can be used for computing trust transitivity in compliance with the requirements described in the previous section. In this section we describes how trust can be expressed and derived with belief calculus, and finally give a numerical example.

9.1 Subjective Logic Fundamentals

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

probability is assigned to the union of possible outcomes. Belief calculus is suitable for approximate reasoning in situations where there is more or less uncertainty about whether a given proposition is true or false.

Subjective logic[11] represents a specific belief calculus that uses a belief metric called *opinion* to express beliefs. An opinion denoted by $\omega_x^A = (b_x^A, d_x^A, u_x^A, a_x^A)$ expresses the relying party A's belief in the truth of statement x. Here b, d, and u represent belief, disbelief and uncertainty, and relative atomicity respectively where $b_x^A, d_x^A, u_x^A, a_x^A \in [0, 1]$ and the following equation holds:

$$b_x^A + d_x^A + u_x^A = 1 \ . \tag{11}$$

The parameter a_x^A reflects the size of the state space from which the statement x is taken. In most cases the state space is binary, in which case $a_x^A = 0.5$. The relative atomicity is used for computing an opinion's probability expectation value expressed by:

$$\mathrm{E}(\omega_x^A) = b_x^A + a_x^A u_x^A \ , \tag{12}$$

meaning that a determines how uncertainty shall contribute to $\mathrm{E}(\omega_x^A)$. When the statement x for example says *"Party B is honest and reliable"* then the opinion can be interpreted as trust in B, which can also be denoted as ω_B^A.

The opinion notation ω_B^A can be used to represent trust relationships, where A and B are the trust originator and target respectively, and (A, B) is the trust edge. The opinion notation normally represents trust relationships as combination of vertexes rather than edges. While the edge and vertex notations are equivalent, their difference is that vertex notation is the most compact, and edge notation is the most explicit because it corresponds directly to algebraic expressions. The connector symbols ":" and "$\diamond$" can be used in both edge and vertex notation.

For example the following trust network in vertex notation:

$$A : E = ((A : B : D) \diamond (A : C : D)) : E \tag{13}$$

is equivalent to Eq.(7) in edge notation and represents the trust network of Fig.5.

The opinion space can be mapped into the interior of an equal-sided triangle, where, for an opinion $\omega_x = (b_x, d_x, u_x, a_x)$, the three parameters b_x, d_x and u_x determine the position of the point in the triangle representing the opinion. Fig.8 illustrates an example where the opinion about a proposition x from a binary frame of discernment has the value $\omega_x = (0.7,\ 0.1,\ 0.2,\ 0.5)$.

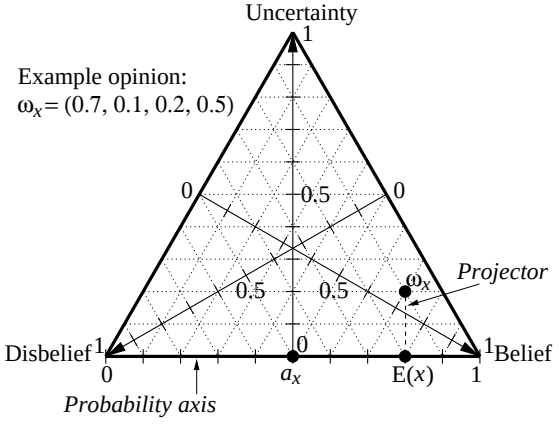

Figure 8: Opinion triangle with example opinion

The top vertex of the triangle represents uncertainty, the bottom left vertex represents disbelief, and the bottom right vertex represents belief. The parameter b_x is the value of a linear function on the triangle which takes value 0 on the edge which joins the uncertainty and disbelief vertices and takes value 1 at the belief vertex. In other words, b_x is equal to the quotient when the perpendicular distance between the opinion point and the edge joining the uncertainty and disbelief vertices is divided by the perpendicular distance between the belief vertex and the same edge. The parameters d_x and u_x are determined similarly. The edge joining the disbelief and belief vertices is called the probability axis. The relative atomicity is indicated by a point on the probability axis, and the projector starting from the opinion point is parallel to the line that joins the uncertainty vertex and the relative atomicity point on the probability axis. The point at which the projector meets the probability axis determines the expectation value of the opinion, *i.e.* it coincides with the point corresponding to expectation value $b_x + a_x u_x$.

Opinions can be ordered according to probability expectation value, but additional criteria are needed in case of equal probability expectation values. We will use the following rules to determine the order of opinions[11]:

Let ω_x and ω_y be two opinions. They can be ordered according to the following rules by priority:

1. The opinion with the greatest probability expectation is the greatest opinion.
2. The opinion with the least uncertainty is the greatest opinion.

Opinions can be expressed as beta PDFs (probability density functions). The beta-family of distributions is a continuous family of distribution functions indexed by the two parameters α and β. The beta PDF denoted by $\mathrm{beta}(\alpha, \beta)$ can be expressed using the gamma function Γ as:

$$\mathrm{beta}(\alpha, \beta) = \frac{\Gamma(\alpha + \beta)}{\Gamma(\alpha)\Gamma(\beta)} p^{\alpha-1}(1-p)^{\beta-1} \tag{14}$$

where $0 \leq p \leq 1$ and $\alpha, \beta > 0$, with the restriction that the probability variable $p \neq 0$ if $\alpha < 1$, and $p \neq 1$ if $\beta < 1$. The probability expectation value of the beta distribution is given by:

$$\mathrm{E}(p) = \alpha/(\alpha + \beta). \tag{15}$$

The following mapping defines how opinions can be represented as beta PDFs.

$$\begin{aligned} &(b_x, d_x, u_x, a_x) \\ &\longmapsto \mathrm{beta}\left(\tfrac{2b_x}{u_x} + 2a_x,\ \ \tfrac{2d_x}{u_x} + 2(1 - a_x)\right). \end{aligned} \tag{16}$$

This means for example that an opinion with $u_x = 1$ and $a_x = 0.5$ which maps to $\mathrm{beta}\,(1, 1)$ is equivalent to a uniform PDF. It also means that a dogmatic opinion with $u_x = 0$ which maps to $\mathrm{beta}\,(b_x\eta,\ d_x\eta)$ where $\eta \to \infty$ is equivalent to a spike PDF with infinitesimal width and infinite height. Dogmatic opinions can thus be interpreted as being based on an infinite amount of evidence.

When nothing is known, the *a priori* distribution is the uniform beta with $\alpha = 1$ and $\beta = 1$ illustrated in Fig.9.

Then after r positive and s negative observations the *a posteriori* distribution is the beta PDF with the parameters $\alpha = r + 1$ and $\beta = s + 1$. For example the beta PDF after observing 7 positive and 1 negative outcomes is illustrated in Fig.10. This corresponds to the opinion of Fig.8 through the mapping of Eq.(16).

A PDF of this type expresses the uncertain probability that a process will produce positive outcome during future observations. The probability expectation value of Fig.10. is $\mathrm{E}(p) = 0.8$. This can be interpreted as saying that the relative frequency of a positive outcome in the future is somewhat uncertain, and that the most likely value is 0.8.

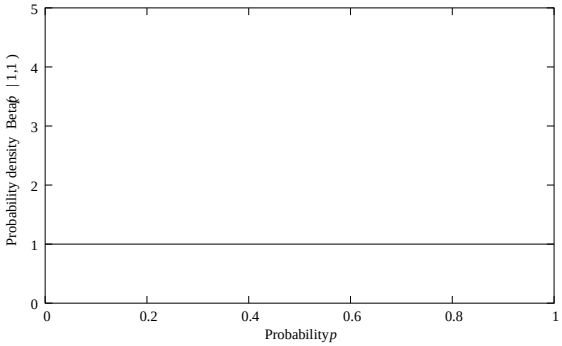

Figure 9: Uniform beta PDF: beta(1,1)

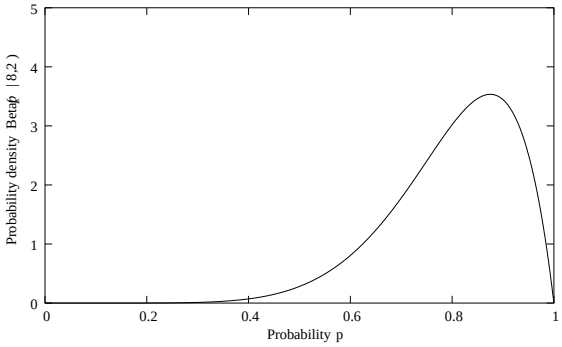

Figure 10: Example beta PDF: beta(8,2)

The variable p in Eq.(14) is a probability variable, so that for a given p the probability density $\mathrm{beta}(\alpha, \beta)$ represents second order probability. The first-order variable p represents the probability of an event, whereas the density $\mathrm{beta}(\alpha, \beta)$ represents the probability that the first-order variable has a specific value.

By definition, the expectation value of the PDF is always equal to the expectation value of the corresponding opinion. This provides a sound mathematical basis for combining opinions using Bayesian updating of beta PDFs.

9.2 Trust Reasoning with Subjective Logic

Subjective logic defines a number of operators[11, 14, 17]. Some operators represent generalisations of binary logic and probability calculus whereas others are unique to belief theory because they depend on belief ownership. Here we will only focus on the *discounting* and the *consensus* operators. The discounting operator can be used to derive trust from transitive trust paths, and the consensus operator can be used to combine parallel transitive trust paths. These operators are described below.

- **Discounting** is used to compute trust transitivity. Assume two agents A and B where A has referral trust in B for the purpose of judging the truth of proposition x denoted by $\omega_B^A = (b_B^A, d_B^A, u_B^A, a_B^A)$. In addition B has function trust in the truth of proposition x, denoted by $\omega_x^B = (b_x^B, d_x^B, u_x^B, a_x^B)$. Agent A can then derive her trust in x by discounting B's trust in x with A's trust in B, denoted by $\omega_x^{A:B} = (b_x^{A:B}, d_x^{A:B}, u_x^{A:B}, a_x^{A:B})$. By using the symbol '$\otimes$' to designate this operator, we can write:

$$\omega_x^{A:B} = \omega_B^A \otimes \omega_x^B \tag{17}$$

where the derived opinion $\omega_x^{A:B}$ is defined by:

$$\begin{cases} b_x^{A:B} = b_B^A b_x^B \\ d_x^{A:B} = b_B^A d_x^B \\ u_x^{A:B} = d_B^A + u_B^A + b_B^A d_x^B \\ a_x^{A:B} = a_x^B \ . \end{cases} \tag{18}$$

The effect of discounting in a transitive path is that uncertainty increases (and not disbelief) [12].

- **Consensus** is equivalent to statistical Bayesian updating. The consensus of two possibly conflicting opinions is an opinion that reflects both opinions in a fair and equal way. Let $\omega_x^A = (b_x^A, d_x^A, u_x^A, a_x^A)$ and $\omega_x^B = (b_x^B, d_x^B, u_x^B, a_x^B)$ be A's and B's opinions about the same proposition x. The opinion $\omega_x^{A\diamond B} = (b_x^{A\diamond B}, d_x^{A\diamond B}, u_x^{A\diamond B}, a_x^{A\diamond B})$ is then called the consensus between ω_x^A and ω_x^B, denoting an imaginary agent $[A, B]$'s opinion about x, as if she represented both A and B. By using the symbol '$\oplus$' to designate this operator, we can write:

$$\omega_x^{A\diamond B} = \omega_x^A \oplus \omega_x^B \ , \tag{19}$$

where the derived consensus opinion is defined by:

$$\begin{cases} b_x^{A\diamond B} = \frac{b_x^A u_x^B + b_x^B u_x^A}{u_x^A + u_x^B - u_x^A u_x^B} \\ d_x^{A\diamond B} = \frac{d_x^A u_x^B + d_x^B u_x^A}{u_x^A + u_x^B - u_x^A u_x^B} \\ u_x^{A\diamond B} = \frac{u_x^A u_x^B}{u_x^A + u_x^B - u_x^A u_x^B} \\ a_x^{A\diamond B} = a_x^A \end{cases} \tag{20}$$

where it is assumed that $a_x^A = a_x^B$. Limits can be computed [13] for $u_x^A = u_x^B = 0$. The effect of the consensus operator is to amplify belief and disbelief and reduce uncertainty.

The discounting and consensus operators will be used for the purpose of deriving trust measures in the example below. Demonstrators for subjective logic operators and trust derivation are available online at: http://security.dstc.com/spectrum/trustengine/ .

9.3 Example Derivation of Trust Measures

This numerical example is based the trust network of Fig.5. Table 2 specifies trust measures expressed as opinions. The DSTC Subjective Logic API[1] was used to compute the derived trust values.

By applying the discounting and consensus operators to the expression of Eq.(8), the derived indirect trust measure can be computed.

- Case a:

  First assume that A derives her trust in E on 31.01.2005, in which case the first entry for (A, B) in Table 2 is used. The expression for the derived trust measure and the numerical result is given below.

$$\begin{aligned} \omega_E^A &= ((\omega_B^A \otimes \omega_D^B) \oplus (\omega_C^A \otimes \omega_D^C)) \otimes \omega_E^D \\ &= (0.74,\ 0.00,\ 0.26,\ 0.50) \end{aligned} \tag{21}$$

[1] Available at http://security.dstc.com/spectrum/

Table 2: Example direct trust measures with reference to Fig.5

Trust Edge	Purpose Variant	Trust Measure	Time
(A, B)	r	$\omega_B^A = (0.9,\ 0.0,\ 0.1,\ 0.5)$	$\tau_1 = 31.01.2005$
(A, C)	r	$\omega_C^A = (0.9,\ 0.0,\ 0.1,\ 0.5)$	$\tau_1 = 31.01.2005$
(B, D)	r	$\omega_D^B = (0.9,\ 0.0,\ 0.1,\ 0.5)$	$\tau_1 = 31.01.2005$
(C, D)	r	$\omega_D^C = (0.3,\ 0.0,\ 0.7,\ 0.5)$	$\tau_1 = 31.01.2005$
(D, E)	f	$\omega_E^D = (0.9,\ 0.0,\ 0.1,\ 0.5)$	$\tau_1 = 31.01.2005$
(A, B)	r	$\omega_B^{'A} = (0.0,\ 0.9,\ 0.1,\ 0.5)$	$\tau_2 = 01.02.2005$

with probability expectation value:

$$\mathrm{E}(\omega_E^A) = 0.87\ . \tag{22}$$

- Case b:

  Let us now assume that, based on new experience on 01.02.2005, A's trust in B suddenly is reduced to that of the last entry for (A, B) in Table 2. As a result of this A needs to update her derived trust in E and computes:

$$\begin{aligned} \omega_E^{'A} &= ((\omega_B^{'A} \otimes \omega_D^B) \oplus (\omega_C^A \otimes \omega_D^C)) \otimes \omega_E^D \\ &= (0.287,\ 0.000,\ 0.713,\ 0.500) \end{aligned} \tag{23}$$

  with probability expectation value

$$\mathrm{E}(\omega_E^{'A}) = 0.64\ . \tag{24}$$

The derived trust measures can be translated into beta PDFs according to Eq.(16) and visualised as density functions as illustrated by Fig.11 and Fig.12 below.

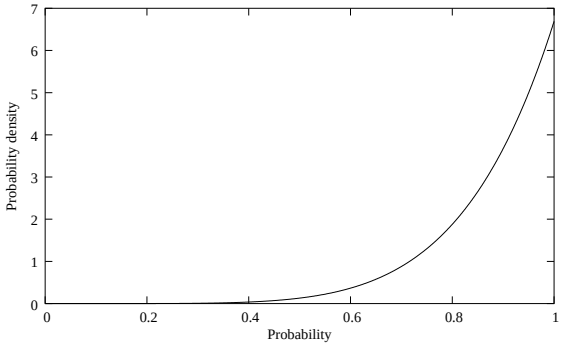

Figure 11: Case a) derived trust: $\omega_E^A \equiv \mathrm{beta}(6.7,\ 1.0)$

It can be seen that the trust illustrated in Fig.11 is relatively strong but that the trust in Fig.12 approaches the uniform distribution of Fig.9, and therefore is very uncertain. The interpretation of this is that the distrust introduced in the (A, B) edge in case b) has rendered the path $A : B : D : E$ (in vertex notation) useless, i.e. when A distrusts B, then whatever B recommends is completely discounted. It is as if B had not recommended anything at all. As a result, A's derived trust in E must be based on the path $A : C : D : E$ (in vertex notation) which was already weak from the start. Thus distrust in a referral does not cause derived distrust in the final functional purpose, but rather the derived trust in the final purpose to be more uncertain.

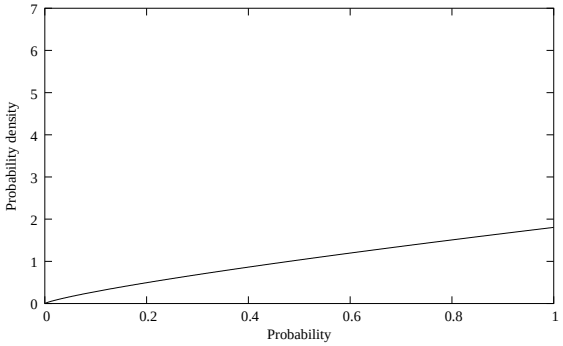

Figure 12: Case b) derived trust: $\omega_E^{'A} \equiv \mathrm{beta}(1.8,\ 1.0)$

The only way to pass distrust intact through a transitive path is when the last trust edge is negative (i.e. distrust) and all the other trust edges in the path are positive. Thus, distrust needs trust in order to propagate through transitive trust networks.

10 Conclusion

We have described principles for modelling trust transitivity that utilise elements of graph theory with additional semantic elements and connectors. This can be expressed with a structured notation that allows concise representation of transitive trust networks.

We described requirements that need to be satisfied in order to make trust transitivity valid. Firstly it is necessary to have matching and semantically consistent trust purposes along transitive trust paths. In particular, we showed that every edge in the path must have the same trust purpose where the last edge has the functional variant of the trust purpose and all previous edges have the referral variant of the trust purpose.

Secondly it is a safety requirement that only direct trust based on personal experience and first hand evidence should be communicated in a trust referral, as communicating indirect trust based on second hand evidence can result in incorrect trust derivation.

We described how integrity and authenticity of referrals in transitive trust paths can be protected by overlaying networks of authentication such as PKIs.

Without referring to any particular algebra or computational methods, we presented several requirements for expressing and computing trust measures. We showed that

belief calculus is a possible candidate for computing transitive trust.

Finally we described a practical example of applying belief calculus for deriving transitive trust using the belief operators of subjective logic, accompanied with visualisation of the derived trust measures using probability density functions.

References

[1] A. Abdul-Rahman and S. Hailes. A Distributed Trust Model. In *Proceedings of the 1997 New Security Paradigms Workshop*, pages 48–60. ACM, 1997.

[2] T. Beth, M. Borcherding, and B. Klein. Valuation of Trust in Open Networks. In D. Gollmann, editor, *ESORICS 94*, Brighton, UK, November 1994.

[3] V. Cahill, B. Shand, E. Gray, et al. Using Trust for Secure Collaboration in Uncertain Environments. *Pervasive Computing*, 2(3):52–61, July-September 2003.

[4] M. Carbone, M. Nielsen, and V. Sassone. A Formal Model for Trust in Dynamic Networks. In *Proc. of International Conference on Software Engineering and Formal Methods (SEFM'03)*, Brisbane, September 2003.

[5] B. Christianson and W. S. Harbison. Why Isn't Trust Transitive? In *Proceedings of the Security Protocols International Workshop*. University of Cambridge, 1996.

[6] R. Falcone and C. Castelfranchi. *Social Trust: A Cognitive Approach*, pages 55–99. Kluwer, 2001.

[7] D. Gambetta. Can We Trust Trust? In D. Gambetta, editor, *Trust: Making and Breaking Cooperative Relations*, pages 213–238. Basil Blackwell. Oxford, 1990.

[8] E. Gray, P. O'Connell, C. Jensen, S. Weber, J.-M. Seigneur, and C. Yong. Towards a Framework for Assessing Trust-Based Admission Control in Collaborative Ad Hoc Applications. Technical Report 66, Dept. of Computer Science, Trinity College Dublin,, dec 2002.

[9] A. Jøsang. The right type of trust for distributed systems. In C. Meadows, editor, *Proc. of the 1996 New Security Paradigms Workshop*. ACM, 1996.

[10] A. Jøsang. An Algebra for Assessing Trust in Certification Chains. In J. Kochmar, editor, *Proceedings of the Network and Distributed Systems Security Symposium (NDSS'99)*. The Internet Society, 1999.

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

[12] A. Jøsang. Subjective Evidential Reasoning. In *Proceedings of the International Conference on Information Processing and Management of Uncertainty (IPMU2002)*, Annecy, France, July 2002.

[13] A. Jøsang, M. Daniel, and P. Vannoorenberghe. Strategies for Combining Conflicting Dogmatic Beliefs. In Xuezhi Wang, editor, *Proceedings of the 6th International Conference on Information Fusion*, 2003.

[14] A. Jøsang and T. Grandison. Conditional Inference in Subjective Logic. In Xuezhi Wang, editor, *Proceedings of the 6th International Conference on Information Fusion*, 2003.

[15] A. Jøsang, S. Hird, and E. Faccer. Simulating the Effect of Reputation Systems on e-Markets. In Nikolau C., editor, *The proceedings of the First International Conference on Trust Management*, Crete, May 2003.

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

[17] A. Jøsang and D. McAnally. Multiplication and Comultiplication of Beliefs. *International Journal of Approximate Reasoning*, 38(1):19–51, 2004.

[18] M. Kinateder and K. Rothermel. Architecture and Algorithms for a Distributed Reputation System. In P. Nixon and S. Terzis, editors, *Proc. of the First International Conference on Trust Management*, number 2692 in LNCS, pages 1–16, Crete, Greece, May 2003. Springer-Verlag.

[19] R. Kohlas and U. Maurer. Confidence valuation in a public-key infrastructure based on uncertain evidence. In *Proceedings of the International Workshop on Theory and Practice of Public-Key Cryptography*. Springer, 2000.

[20] D.W. Manchala. Trust Metrics, Models and Protocols for Electronic Commerce Transactions. In *Proceedings of the 18th International Conference on Distributed Computing Systems*, 1998.

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

[22] D.H. McKnight and N.L. Chervany. The Meanings of Trust. Technical Report MISRC Working Paper Series 96-04, University of Minnesota, Management Information Systems Reseach Center, URL: http://misrc.umn.edu/wpaper/, 1996.

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

[24] P Yolum and M.P) Singh. Dynamic Communities in Referral Networks. *Web Intelligence and Agent Systems (WIAS)*, 1(2):105–116., 2003.

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