Analysing Topologies of Transitive Trust

Audun Jøsang[1], Elizabeth Gray[2], and Michael Kinateder[3]

[1] Distributed Systems Technology Centre[†]
Queensland University of Technology, Brisbane, Australia
Email: ajosang@dstc.edu.au
[2] Computer Science Department[‡]
Trinity College Dublin, Ireland
Email: Liz.Gray@cs.tcd.ie
[3] Institute of Parallel and Distributed Systems [§]
Faculty of Computer Science, University of Stuttgart, Germany
Email: Michael.Kinateder@informatik.uni-stuttgart.de

Abstract. Transacting and interacting through computer networks makes it difficult to use traditional methods for establishing trust between parties. Creating substitutes by which people, organisations and software agents can derive trust in others through computer networks requires computerised analysis of trust topologies. This paper describes diverse dimensions of trust that are needed for analysing trust topologies, and provides a notation with which to express trust relationships in terms of these dimensions. The result is a simple way of specifying topologies of trust from which derived trust relationships can be automatically and securely computed.

1 Introduction

Modern communication media are increasingly removing us from familiar styles of interacting and doing business which both rely on some degree of trust between the interaction or business partners. Moreover most traditional cues for assessing trust in the physical world are not available through those media. We may now be conducting business with people and organisations of which we know nothing, and we are faced with the difficult task of making decisions involving risk in such situations. As a result the topic of trust in open computer networks is receiving considerable attention in the network security community and e-commerce industry [1–4]. State of the art technology for stimulating trust in e-commerce includes cryptographic security mechanisms for providing confidentiality of communication and authentication of identities. However, merely having a cryptographically certified identity or knowing that the communication

[†] 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 Industry, Science & Resources).

[‡] This work has been supported in part by the SECURE project (IST-2001-32486), a part of the EU FET Global Computing initiative.

[§] Part of this work has been funded by Hewlett-Packard Limited.

channel is encrypted is not enough for making informed decisions if no other knowledge about a remote transaction partner is available. Trust therefore also applies to for example the reliability, honesty and reputation of transaction partners.

Being able to formally express and reason with these types of trust is needed in order to create substitutes for the methods we use in the physical world, and also for creating new methods for determining 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 a notation for specifying topologies of transitive trust, and to discuss ways to reason about trust in such topologies. We first consider properties of trust: diversity, transitivity, and combination. We then propose a notation for describing and reasoning about trust, and illustrate how this notation may successfully and securely be used to correctly analyse different trust scenarios. Finally, we identify several requirements that trust measures and operators should satisfy.

2 Trust Diversity

Humans use trust to facilitate interaction and accept risk in situations where complete information is unavailable. However, trust is a complex concept that is difficult to stringently define. A wide variety of definitions of trust have been put forward [5], many of which are dependent on the context in which interaction occurs, or on the observer's subjective point of view. Deutsch's definition of trust is commonly used as a starting point for understanding:

> If an individual is confronted with an ambiguous path, a path that can lead to an event perceived to be beneficial (Va^{+}) or to an event perceived to be harmful (Va^{-}); he perceives that the occurrence of (Va^{+}) or (Va^{-}) is contingent on the behaviour of another person; and he perceives that the strength of (Va^{-}) to be greater than the strength of (Va^{+}). If he chooses to take an ambiguous path with such properties, I shall say he makes a trusting choice; if he chooses not to take the path, he makes a distrustful choice. [6]

While Deutsch breaks trust down further into several different circumstances in which a trusting choice might be made, he concentrates on the fact that trust "is strongly liked to confidence in, and overall optimism about, desirable events taking place." [7]

A similar description of trust has been expressed by McKnight and Chervany and can be summarised as follows:

> Trust is the extent to which one party is willing to depend on something or somebody in a given situation with a feeling of relative security, even though negative consequences are possible. [5]

This definition illustrates that non-living material or abstract things can also be trusted although they do not have a free will to behave honestly or dishonestly in the way living persons do. McKnight and Chervany also separate between different trust constructs, including *trusting behaviour* which expresses the act of entry into a situation of dependence, *trusting intention* which is only the intention to do so, and *system trust* which denotes trust in “impersonal structures”, either material or abstract.

Thus, we may say that trust is related to belief in the honesty, reliability, competence, willingness, etc. of the trusted entity, it being a person, organisation, system. Trust can also be related to a particular property of material or abstract objects such as a computer system or our legal institutions. Despite this variation in meanings, many researchers simply use and assume a definition of trust in a very specific way, such as *a trusted public key* which refers to the authenticity of that key.

The repeated uses of the word “perceives” in Deutsch’s definition implies that trust is a subjective quality individuals place in one another. Additionally, the fact that different entities can have different kinds of trust in the same target entity indicates that trust is subjective. It is also important to notice that trust is related to the purpose and nature of the relationship, e.g. an organisation trusts an employee to deal with financial transactions up to a specific amount, but not above, and that same employee might not be trusted to make public statements about the organisation.

In order for trust to form topologies it needs to be expressed with three basic diversity dimensions [8] where the first dimension represents the trustor or trust origin, the second represents the trust purpose, and the third represents the trustee or the trust target. This is illustrated in Fig. 1 below.

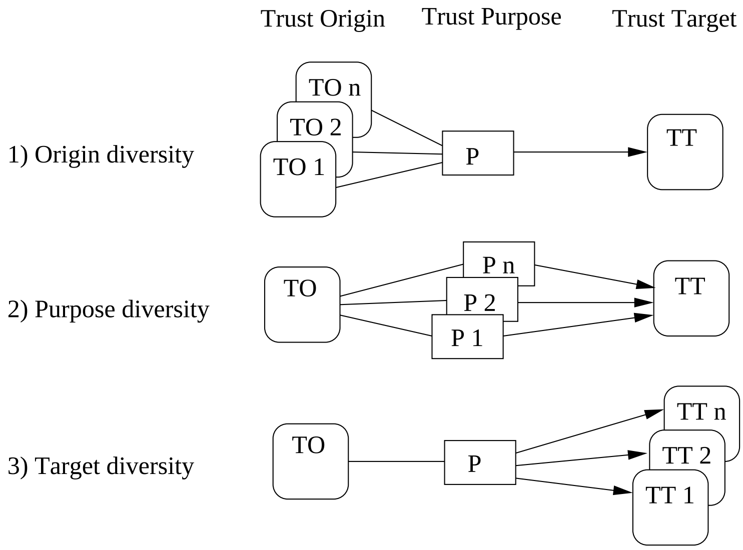

Fig. 1. Basic trust diversity

In addition to the three basic topology dimensions a *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. 7.

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 entities trustor and 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 [9] that trust is not implicitly transitive. However, a recommendation system can be used to allow trust transitivity according to explicit conditions.

Trust transitivity means, for example, that if Alice trusts Bob who trusts Clark then Alice will also trust Clark. This assumes that Bob actually tells Alice that he trusts Clark, and this will typically happen in a recommendation. In this simple example the trust origins and trust targets are easily identifiable, but it does not say anything specific about the trust purposes.

Let us assume that Alice needs to have her car serviced, so she asks Bob for his advice about where to find a good car mechanic in town. Bob is thus trusted by Alice to know about a good car mechanic and to tell his honest opinion about that, whereas Clark is trusted by Bob to be a good car mechanic.

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. The trust origins and targets are again explicit, but it is more tricky to define exactly for what purpose Alice now 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 becomes proportional with the length of the transitive chain, 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 recommend represents indirect trust and is precisely what allows trust to become transitive. At the same time this trust always assumes the existence of a direct trust purpose at the end of the transitive path which in the example above is about being a good car mechanic.

This observation indicates that the trust purpose of the final leg must somehow be part of every leg in the trust path. We will express this by defining the two trust variants *indirect* and *direct* and let the *trust variant* be a parameter in every trust purpose.

Alice would then have *indirect* trust in Bob to be a good car mechanic, and similarly for Bob and Claire. This must be interpreted as saying that Alice trusts Bob to recommend somebody (to recommend somebody etc.) to be a good car mechanic. On the other hand Claire would have *direct* trust in David to be a good car mechanic. The indirect variant of a trust purpose is recursive so that any transitive trust chain, with arbitrary length, can be expressed using only one trust purpose with two variants.

The examples above assume some sort of absolute trust between the agents in the transitive chain. In reality trust is never absolute, and many researchers have proposed to express trust as discrete verbal statements, as probabilities or other continuous measures. 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 trust in the car mechanic David through the recommenders Bob and Claire can be at most as strong as Claire's trust in David. This is illustrated in Fig. 2 below.

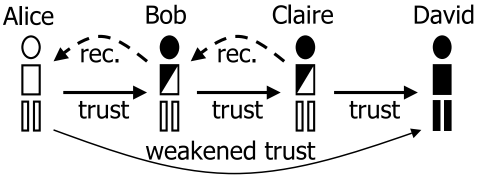

Fig. 2. Trust transitivity

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

It could be argued that negative trust in a transitive chain can have the paradoxical effect of strengthening the derived trust. Take for example the case where Bob distrusts Claire and Claire distrusts David whereas Alice trusts Bob. 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 we will not go into further detail on this issue here.

We will use the symbol ":" to denote initial trust relationships and "$\overset{\rightarrow}{:}$" to denote derived trust relationships, so that the trust relationships of Fig. 2 can be expressed as:

$$\text{Alice} \overset{\rightarrow}{:} \text{David} = \text{Alice} : \text{Bob} : \text{Claire} : \text{David} \tag{1}$$

where the trust purpose is implicit. Let the trust purpose be defined as P_1; " *trusts X to be a good car mechanic* ". Let the direct variant be denoted by dP_1 and the indirect variant by iP_1. The trust topology of Fig.2 can then be explicitly expressed as:

$$\text{Alice}; [dP_1] \overset{\rightarrow}{:} \text{David} = \text{Alice}; [iP_1] : \text{Bob}; [iP_1] : \text{Claire}; [dP_1] : \text{David} \tag{2}$$

The idea of contstructing transitive trust chains based on a single trust purpose with direct and indirect variants is captured by the following definition.

Definition 1. *A valid transitive trust chain requires that every leg in the chain contains the same trust purpose and that every leg except the last is indirect.*

The transitive path stops when a leg is not indirect. It is of course possible for a principal to have both direct and indirect trust in another principal but that should be expressed as two separate trust legs. The existence of both a direct and an indirect trust leg 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 somebody else for the job.

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_3 respectively. Alice's derived trust measure and time stamp are denoted by μ_4 and τ_4. The trust expression of Fig. 2 can then be expressed as:

$$\begin{aligned}\text{Alice}; [dP_1, \mu_4, \tau_4] \overset{\rightarrow}{:} \text{David} = {} & \text{Alice}; [iP_1, \mu_1, \tau_1] : \text{Bob}; [iP_1, \mu_2, \tau_2] \; : \\ & \text{Claire}; [dP_1, \mu_3, \tau_3] : \text{David}\end{aligned} \tag{3}$$

Claire obviously recommends to Bob her opinion about David as a car mechanic, but Bob's recommendation to Alice is ambiguous. It can either be that Bob passes Claire's recommendation unaltered on to Alice, or that Bob derives his own direct trust in David which he recommends to Alice. The latter way of passing recommendations can create problems and it is better when Alice receives Claire's recommendation unaltered. This will be discussed in more detail in Sec. 5.

4 Parallel Trust Combination

It is common to collect recommendations 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 as 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.

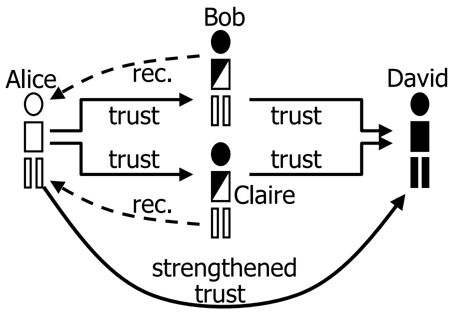

Fig. 3. Parallel trust combination

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

We will use the symbol "," to denote combination of trust, so that Alice's combination of the two parallel trust chains from her to David can be expressed as:

$$\text{Alice} \overset{\rightarrow}{:} \text{David} = (\text{Alice} : \text{Bob} : \text{David}), (\text{Alice} : \text{Claire} : \text{David}) \tag{4}$$

In the above expression the trust purpose is implicit, and the following expression makes it explicit with regard to the trust purpose:

$$\begin{aligned} \text{Alice}; [dP_1] \overset{\rightarrow}{:} \text{David} = {} & (\text{Alice}; [iP_1] : \text{Bob}; [dP_1] : \text{David}), \\ & (\text{Alice}; [iP_1] : \text{Claire}; [dP_1] : \text{David}) \end{aligned} \tag{5}$$

5 Topology Analysis

Trust topologies can involve many principals, and capital letters $A, B, ...$ will be used to denote principals instead of names such as Alice and Bob.

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

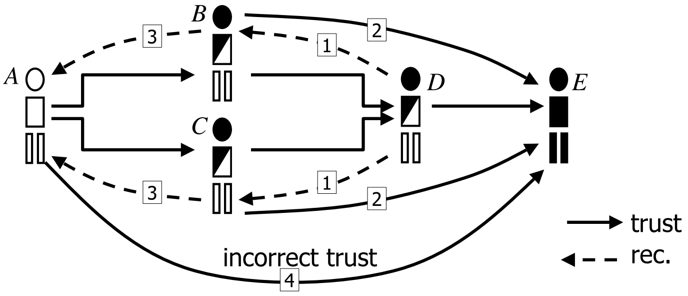

Fig. 4. Incorrect recommendation

In Fig. 4 the trust and recommendation arrows are indexed according to the order in which they are formed whereas the initial trust relationships have no index. In the scenario of Fig.4 D passes his recommendation about E to B and C (index 1) so that B and C are able to derive direct trust in E (index 2). Now B and C pass their derived trust in E to A (index 3) so that she can derive direct trust in E (index 4). As a result A perceives the topology to be $(A : B : E), (A : C : E)$. The problem with this scenario is that A is ignorant about D so that A in fact derives a hidden topology that is different from the perceived topology:

$$\begin{array}{ccc} \text{Perceived topology:} & & \text{Hidden topology:} \\ & & \\ (A : B : E), (A : C : E) & \neq & (A : B : D : E), (A : C : D : E) \end{array} \tag{6}$$

The reason for this is that B's trust $B \overset{\rightarrow}{:} E$ was derived from $B : D : E$ and C's trust $C \overset{\rightarrow}{:} E$ was derived from $C : D : E$, so when B and C recommend E they implicitly recommend $B : D : E$ and $C : D : E$ [10] but this is hidden from A. It can easily be seen that neither the perceived nor the hidden topology is equal to the real topology, which shows that this way of passing recommendations produces incorrect results.

We argue that B and C should pass the recommendations explicitly as $B : D : E$ and $C : D : E$ respectively, and this is certainly possible, but then A needs to be convinced that B and C have not altered the recommendations from D. If B and C are unreliable they might for example try to change the recommended trust measures. Not only that, any party that is able to intercept the recommendations from B, C, or D to A might want to alter the trust values, and A needs to receive evidence of the authenticity and integrity of the recommendations. Cryptographic security mechanisms can typically be used to solve this problem, and this will be discussed in more detail in Sec.6.

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

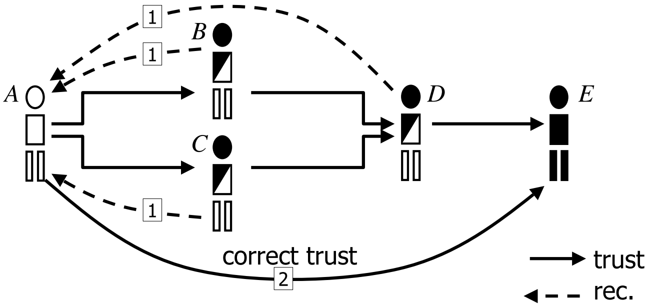

Fig. 5. Correct recommendation

In the scenario of Fig. 5 the perceived topology is equal to the real topology which can be expressed as:

$$A \overset{\rightarrow}{:} E = ((A : B : D), (A : C : D)) : E \tag{7}$$

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 recommendations from others. We will use the term *primary trust* to denote the former, and *secondary trust* for the latter. Fig. 4 illustrated how problems can occur when secondary trust is recommended, so the rule is to only recommend primary trust [10]. Derived trust is per definition always secondary trust so that for example, A's derived trust in E in Fig. 5 should not be recommended to others.

Expressing transitive chains in the form of Eq. (3) is not always practical, for example when the topology is large, or when only parts of it are known. Instead, each isolated trust relationship can be expressed individually, and an automated parser can establish valid topologies depending on the need.

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

Table 1. Initial trust relationships of Fig.5

Origin	Target	Purpose	Variant	Measure	Time
A	B	P_1	indirect	μ_1	$\tau_3 = 08.09.2003$
A	C	P_1	indirect	μ_2	$\tau_3 = 08.09.2003$
B	D	P_1	indirect	μ_3	$\tau_1 = 03.09.2003$
C	D	P_1	indirect	μ_4	$\tau_1 = 03.09.2003$
D	E	P_1	direct	μ_5	$\tau_1 = 03.09.2003$
D	E	P_1	direct	μ_6	$\tau_2 = 05.09.2003$

A parser going through Table 1 will be able to determine the topology of Fig. 5. The principal A can be called a relying party because she relies on the recommendations 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 recommendations. In Table 1 it can be observed that there are two entries for the trust leg $D : E$ and based on the principle of the most recent trust, the parser would select the last entry expressed by $D; [dP_1, \mu_6, \tau_2] : E$. If the relying party A derives her trust in E at time τ_3, then that trust can be expressed as:

$$\begin{aligned} A; [dP_1, \mu_7, \tau_3] \overset{\rightarrow}{:} E = ((&A; [iP_1, \mu_1, \tau_3] : B; [iP_1, \mu_3, \tau_1] : D), \\ &(A; [iP_1, \mu_2, \tau_3] : C; [iP_1, \mu_4, \tau_1] : D)); [dP_1, \mu_6, \tau_2] : E \end{aligned} \quad (8)$$

The piece of pseudo-code below represents a parsing algorithm that finds a trust path for a specific origin, target and trust purpose, if it exists, in a set of recommendations. It evaluates all possible trust paths as true or false, and uses binary logic OR to combine parallel trust paths. This simplification assumes that trust measures are binary. As already mentioned, trust can be measured as discrete or continuous values, in which case a more complex algorithm would be needed.

This simple algorithm can be useful to determine if at least one potential trust path exists between two principals, and further analysis can then be done to derive the measure of trust resulting from the topology. The latter must be based on algebraic operators for computing transitive and parallel trust. This issue will be briefly discussed in Sec 7.

```
Pseudo-Constructor for a Recommendation:
========================================

Recommendation(Principal origin, Principal target, Purpose purpose,
               Variant variant) {
  this.origin = origin;
  this.target = target;
  this.purpose = purpose;
  this.variant = variant;
}
```

```
Pseudo-code for a simple evaluation algorithm:
==============================================

Output is binary:
  true  --> there is a trust path
  false --> there is none

Definition of functions:
  transitivity       : --> logical AND
  trust combination  , --> logical OR

boolean ParseTrust(Principal origin, Principal target, Purpose purpose,
                   RecommendationSet recs) {
  IF ((origin, target, purpose, ‘direct’) IN recs) {
    RETURN true;
  }
  ELSE {
    SELECT rec FROM recs WHERE ((rec.origin == origin) AND
                                (rec.purpose == purpose) AND
                                (rec.variant == ‘indirect’));
    IF (RESULTS_FROM_SELECT == empty) {
      RETURN false;
    }
    ELSE {
      Boolean b = false;
      FOREACH rec IN RESULTS_FROM_SELECT DO {
        b = b OR ParseTrust(rec.target, target, purpose, recs\rec);
      }
      RETURN b;
    }
  }
}
```

6 Integrity and Authenticity of Recommendations

Cryptography can be used to provide authenticity and integrity of recommendations. 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 create trust management problems. 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 chain 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.

The concept of trusted platforms introduces additional security features to reputation systems in general, and uses cryptographic means to secure recommendations and trust assessments in particular [11]. 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 recommendations communicated can be assumed, and trust topologies 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 topology in question 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] and it broadly follows the same principles as described in the previous sections.

The consequence of having to derive trust in public keys is that the relying party might have to analyse a separate topology for every principal in the topology of interest. The analysis of the topology of Fig.5 which includes 3 recommendations would for example require the derivation of the trust in the public keys of B, D and C before the topology itself can be analysed and the trust in the target entity E can be derived. The analysis of the topology would then have to take the authenticity of the public keys into account in addition to the trust in the principals. With reference to the scenario of Fig.5 the trust relationships that have to be taken into account are illustrated in Fig.6 below.

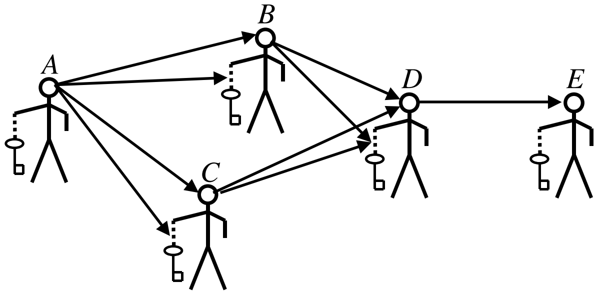

Fig. 6. Trust relationships in a topology with authenticated public keys

The existence of two separate trust legs with different purposes, where the first is targeted at the principal himself and the second at the binding between the public key and its owner requires some method for combining the two 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 purpose of the trust targeted at the public key-to-owner binding can typically be described as P_2; " *trusts the public key to be authentic* ", and it can be associated with a measure and timestamp as in the normal case. We will use the symbol "$\wedge$" to denote conjunction of two trust relationships. A's derived trust in E can then be expressed as:

$$
\begin{aligned}
&A;[dP_1,\mu_7,\tau_3]\overset{\rightarrow}{:}E \;\;= \\
&(\;\;(A;[iP_1,\mu_1,\tau_3]\wedge[dP_2,\mu_8,\tau_3]:B;[iP_1,\mu_3,\tau_1]\wedge[dP_2,\mu_9,\tau_1]:D), \\
&\quad(A;[iP_1,\mu_2,\tau_3]\wedge[dP_2,\mu_{10},\tau_3]:C;[iP_1,\mu_4,\tau_1]\wedge[dP_2,\mu_{11},\tau_1]:D)\;); \\
&[dP_1,\mu_6,\tau_2]:E
\end{aligned}
\tag{9}
$$

For a parser to be able to derive this topology, it is of course required that the relying party A has received and stored all these trust recommendations 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 aactual topology. 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 result of the conjunction between trust in the principal and the distrust in his key would produce distrust, indicating that a principal identified by this particular public key can not be trusted. This is what intuition would dictate because it is now possible that recommendations that appear to come from the principal in fact originate from the intruder who stole the private key and who is not trusted.

7 Measuring and Computing Trust

In previous sections we have used the term "trust measure" without specifying how it should be expressed or computed, and that is not the topic of this paper. Instead, we have indicated several intuitive principles that trust measures and computational rules should follow. Without going into great detail this section describes additional requirements that trust measures and operators should satisfy.

While trust has no specific measurable units, its value can be measured in a similar manner to other abstract commodities, like information or knowledge [12]. Many trust measures have been proposed in the literature varying from discrete measures [13–17] to continuous measures [10, 18–23].

Typical discrete trust measures are for example "strong trust", "weak trust", strong distrust" and "weak distrust". PGP[13] 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 origin and the target transacting partners. Otherwise, if trust is subjectively measured by each party using different methods, 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 so that they are deriving meaningful trust values for one another.

The *context*, or purpose, of the interaction must also be satisfied by the trust measure. Again, by explicitly defining P_1, and P_2, the context becomes clear to all parties participating in the interaction.

As mentioned in Sec. 2, *time* is another 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 chains should result in a reduced confidence level. On the other hand, a large number of parallel recommendations should result in an increased confidence level.

Finally, in order to derive trust measures from a topology there must be explicit methods for *combining trust measures in a transitive chain* as in Fig.2, for *combining trust measures in parallel chains* as in Fig.3 as well as for *combining trust measures in a conjunction of trust relationships* as in Fig.6. Various methods and principles for deriving trust from such combinations have been proposed in the literature [10, 13, 14, 16, 18–20]. 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.

8 Conclusion

We have captured the diversity that exists in trust by specifying three basic topology dimensions, that of trust origin, trust target and trust purpose. Additionally, we have incorporated the dimensions of measure and time into the specification which are important for deriving trust measures through computational methods.

We have described principles for recommendation such that transitive trust chains might be formed which capture the basic trust diversity dimensions. In this regard, we found that a trust topology is valid when every leg in the chain contains the same trust purpose with the last leg having direct trust and all previous legs having indirect trust.

We provided a notation with which to express these trust principles and to analyse topologies of transitive trust. In doing so, we proved the rule that only primary trust should be recommended, as recommending secondary trust results in incorrect trust derivation.

We showed also that the parsing of transitive trust chains may be automated such that trust measures might be derived practically and easily in scenarios where the topology is large or where only parts of the topology are known. We presented the pseudocode for such a parser.

Finally, we presented a method with which to ensure the integrity and authenticity of recommendations in transitive trust chains, as well as several requirements for expressing and computing trust measures.

References

1. Blaze, M., Feigenbaum, J., Lacy, J.: Decentralized Trust Management. In: Proc. of the 17th IEEE Symposium on Security and Privacy, Oakland (1996) 164–173
2. Blaze, M., Feigenbaum, J., Ioannidis, J., Keromytis, A.D.: The KeyNote Trust-Management System. RFC 2704, Network Working Group, IETF (1999)
3. Ellison, C., Frantz, B., Lampson, B., Rivest, R.L., Thomas, B., Ylonen, T.: SPKI Certificate Theory. RFC 2693, Network Working Group, IETF (1999)
4. Rivest, R., Lampson, B.: SDSI - A Simple Distributed Security Infrastructure (1996)
5. McKnight, D., Chervany, N.: The Meanings of Trust. Technical Report MISRC 96-04, Management Informations Systems Research Center, University of Minnesota, (1996)
6. Deutsch, M.: Cooperation and Trust: Some Theoretical Notes. In Jones, M., ed.: Nebraska Symposium on Motivation, Nebraska University Press (1962)
7. Golembiewski, R., McConkie, M.: 7. In: The Centrality of Interpersonal Trust in Group Processes. Wiley (1975) 131–185
8. Jøsang, A.: The right type of trust for distributed systems. In Meadows, C., ed.: Proc. of the 1996 New Security Paradigms Workshop, ACM (1996)
9. Christianson, B., Harbison, W.: Why Isn't Trust Transitive? In: Proc. of the Security Protocols Workshop. (1996) 171–176
10. Jøsang, A.: An Algebra for Assessing Trust in Certification Chains. In Kochmar, J., ed.: Proc. of the Network and Distributed Systems Security Symposium (NDSS'99), The Internet Society (1999)
11. Kinateder, M., Pearson, S.: A Privacy-Enhanced Peer-to-Peer Reputation System. In: Proc. of the 4th International Conference on Electronic Commerce and Web Technologies (EC-Web 2003), Prague, Czech Republic, Springer-Verlag (2003)
12. Dasgupta, P.: Trust as a Commodity (2000)
13. Zimmermann, P.: The Official PGP User's Guide. MIT Press (1995)
14. Abdul-Rahman, A., Hailes, S.: A Distributed Trust Model. In: Proceedings of the 1997 New Security Paradigms Workshop, ACM (1997) 48–60
15. Cahill, V., Shand, B., Gray, E., et al.: Using trust for secure collaboration in uncertain environments. To appear in IEEE Pervasive Computing (2003)
16. Carbone, M., Nielsen, M., Sassone, V.: A formal model for trust in dynamic networks. Technical Report RS-03-4, BRICS (2003)
17. Manchala, D.: Trust Metrics, Models and Protocols for Electronic Commerce Transactions. In: Proceedings of the 18th International Conference on Distributed Computing Systems. (1998)
18. Jøsang, A.: A Logic for Uncertain Probabilities. International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems **9** (2001) 279–311
19. Kohlas, R., Maurer, U.: 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. Kinateder, M., Rothermel, K.: Architecture and Algorithms for a Distributed Reputation System. In Nixon, P., Terzis, S., eds.: Proc. of the First International Conference on Trust Management. Number 2692 in LNCS, Crete, Greece, iTrust, Springer-Verlag (2003) 1–16
21. Gray, E., O'Connell, P., Jensen, C., Weber, S., Seigneur, J.M., Yong, C.: Towards a Framework for Assessing Trust-Based Admission Control in Collaborative Ad Hoc Applications. Technical Report 66, Dept. of Computer Science, Trinity College Dublin, (2002)
22. Beth, T., Borcherding, M., Klein, B.: Valuation of Trust in Open Networks. In Gollmann, D., ed.: ESORICS 94, Brighton, UK (1994)
23. Marsh, S.: Formalising Trust as a Computational Concept. Phd thesis, University of Stirling, Department of Computer Science and Mathematics (1994)