

A Calculus of Trust and Its Application to PKI and Identity Management

Jingwei Huang
Information Trust Institute
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801, USA
jingwei@iti.illinois.edu

David Nicol
Information Trust Institute
Dept. of Electrical & Computer Engineering
University of Illinois at Urbana-Champaign
1308 West Main Street
Urbana, Illinois 61801, USA
nicol@iti.illinois.edu

ABSTRACT

We introduce a formal semantics based calculus of trust that explicitly represents trust and quantifies the risk associated with trust in public key infrastructure (PKI) and identity management (IdM). We then show by example how to formally represent trust relationships and quantitatively evaluate the risk associated with trust in public key certificate chains. In the context of choosing a certificate chain, our research shows that the shortest chain need not be the most trustworthy, and that it may make sense to compare the trustworthiness of a potential chain against a threshold to govern acceptance, changing the problem to finding a chain with sufficiently high trustworthiness. Our calculus also shows how quantified trust relationships among CAs can be combined to achieve an overall trust assessment of an offered certificate.

Categories and Subject Descriptors

K.6.5[Management of Computing and Information Systems] [Security and Protection]; I.2.11 [Distributed Artificial Intelligence]

General Terms

Theory, Measurement, Security

Keywords

Trust modeling, PKI, Identity management, Risk assessment, Uncertainty, Semantics of trust, Social networks

1. INTRODUCTION

Trust plays a crucial role and is recognized as a major risk factor in public key infrastructure (PKI) and identity management (IdM). This paper explicitly represents trust with well defined semantics, and quantifies the risk associated with trust in PKI and IdM.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IDTrust '09, April 14-16, 2009, Gaithersburg, MD, USA
Copyright 2009 ACM 978-1-60558-474-4 ...\$5.00.

Trust issues exist throughout identity management mechanisms. As identified in [2], organizations have concern about the business rules and mechanisms of their IdM partners with respect to the use of shared identity and credential information, how their partners protect the information, and the quality of identity information they provide; individuals (identity owners) are concerned whether their identity information in an organization is secure (not being stolen or revealed), how the information is used, and with whom it is shared. These trust concerns are beyond technologies, but are tightly associated with organizational and human behaviors. They are most difficult factors in identity management.

Digital signature and certification, facilitated by PKI, is considered to provide secure communication in the Internet, and as a fundamental tool to support IdM. However, many risks exist in PKI. The number one risk identified by Ellison and Schneier [8], "Who do we trust, and for what", reveals the risk of "imprecise use of the word 'trust' "; to avoid such risk, the use of trust relationships in digital certification and validation need to be precise and to be specific. An incident [10] in which VeriSign issued an impostor two digital certificates associated with Microsoft still reminds people to think whether a certification authority (CA) can be safely trusted regarding the validity of the issued certificates.

A number of PKI trust models have been proposed[33] [23] [30] [37]. However, these studies focus on the relationships among certification authorities (or the structure of PKI), the certification path construction methods, and the performance of different PKI structures and path build methods. Left missing is the explicit and accurate representation of trust relationships and the quantification of risks associated with trust in certification paths.

In PKI, a certification path corresponds to a chain of trust relationships. In distributed IdM such as federated IDM systems, a credential chain also corresponds to a chain of trust relationships. What do these trust relationships mean, exactly? On what precisely does one entity trust another in a credential chain? What is the specific context of each trust? How do we quantify the risk associated with trust in a credential chain?

In a typical public key validation model using PKI, at least one certification path with shortest length needs to be found and validated. When multiple paths exist, typically a short path is chosen in order to minimize the work involved. From the risk point of view this implicitly assumes that each certificate has the same level of risk, so the longer a certifica-

tion path is, the higher the risk is. However, when quantified evaluation of risk is introduced to PKI and different risks become associated with different certificates, some very interesting issues emerge. There are multiple certification paths, but which certification path should be chosen? Should more than one certification path be considered? Should all certificate paths be considered?

This paper explores some answers to these questions by introducing and applying a formal semantics based calculus of trust [17] to explicitly represent trust, and to quantify risk associated with trust in PKI and IdM.

This paper is organized as follows. Section 2 introduces related research; section 3 introduces a motivating scenario; section 4 discusses the semantics of trust; from a probabilistic perspective, section 5 discusses the measurement of uncertain trust; section 6 discusses sequence trust aggregation and parallel trust aggregation models; section 7 applies the trust calculus to formally represent trust relationships and quantitatively evaluate the risk associated with trust in public key certificate chains. Section 8 summarizes the research and discusses future directions.

2. RELATED RESEARCH

In this section, we review the research related to trust models for PKI, trust formalization and quantification.

Trust in PKI.

A number of PKI trust models have been studied [33] [23] [30] [37]. These studies focus on the relation among certification authorities (the structure of PKI), the certification path construction methods, and the performance analysis of different structures and path build methods. However, explicit and accurate representation of trust relationships and the quantification of risks associated with trust in certification paths is not a consideration of that research.

Most of public key certification path construction methods [9] [40] [29] implicitly assume that a certificate has the same level of risk, so a certification path with the shortest length has the least risk of being invalid. However, different certification authorities have different levels of rigor in the identity verification they apply before issuing a certificate, so that different certificates have different levels of risk.

Reiter and Stubblebine [36] work towards a quantitative evaluation of risk by studying the metrics for authentication in PKI. Based on their model of “resilient authentication”, the authors suggested two metrics to measure the risk in public key certification: the maximum number of independent paths with bounded length, and the maximum number of nodes which have to be removed (compromised) to break all certification paths. The proposed metrics are distinguished by their resiliency to malicious behaviors, but still follow the implicit assumption that each certificate has the same level of risk.

Maurer [27] explicitly represent certificate, trust, and recommendation in a certificate chain in PKI, and quantified uncertainty of the validity of a statement as “confidence level” in $[0,1]$. The limitation of his representation is that a single number between 0 and 1 can represent the uncertainty regarding the validity of a statement, but cannot represent the uncertainty due to incomplete knowledge, which is necessary in modeling trust in an open environment.

Trust Formalization and Quantification.

There are two major streams of research on trust formalization: logical approach and quantitative approach. The logical stream mainly focuses on the semantic structure of trust, the logical conditions and effects of trust. Examples include [4] [6] [16].

The quantitative stream mainly focuses on the uncertainty of trust, trust quantification, and the models & algorithms of trust computing. Trust has been quantified in several ways, at least including: linguistic values, graded levels, subjective probability, and probability distribution. In Marsh’s formalism [25], trust is quantified as a number in interval $[-1, +1]$; $+1$ represents complete trust, -1 represents completely distrust, and 0 represents “no trust” (untrust). In this way, Marsh clearly discerns untrust and distrust; but the relation between trust, untrust and distrust is somewhat oversimplified. A trust value is either between trust and untrust or between untrust and distrust. Based on a thorough examination of the concepts of trust developed in social sciences, Marsh designed a set of formulas to express the relations in trust; while Marsh’s model basically is a heuristic formalism. Most quantitative trust models define trust degree as a real value in interval $[0,1]$, e.g. [27] [31] [20]. As addressed in [12], it is a problem regarding how to interpret the meaning of 0 , which could be untrust or distrust. Ding et al. [7] defined 1 as fully trust, 0 as fully distrust, and 0.5 as fully ignorant (untrust). This is a simple approach to discern untrust and distrust. Most models in this school of research are heuristic, the measurement of trust is subjective, and the semantics of trust is not formally defined.

Josang [18] addressed uncertain belief representation by using subjective logic, in which an opinion regarding belief is represented as a triple (b, d, u) , where b , d , and u denote the degrees of belief, disbelief, and uncertainty, respectively, and $b+d+u = 1$. Later, Josang et al. [19] applied the subjective logic to represent uncertain trust. Explicit inclusion of uncertainty u enables to express and explain degrees of trust, distrust, and untrust, which takes into account incomplete knowledge about a trustee. We adopt this trust measure triple, but our model is different from Josang’s model [19]. In their model, although belief opinion is formally modeled, but the semantics of trust is not formally defined, and the relation between belief and trust is not modeled; hence, their trust model was subjectively defined as a heuristic formulation. Although they introduced degree of distrust, their trust derivation failed to clearly discern the semantics of untrust and distrust. We formally define each degree (of trust, distrust, and undecidable) in terms of formal semantics of trust, and based on the formal semantics, we derive trust calculation formulas rather than subjectively define the model.

Social Networks based Trust.

The Web has become an open dynamic and decentralized information/knowledge repository, global electronic markets, and a distributed computing platform. In such a cyberspace, people, organizations and software agents need to interact with others with unknown identity. To meet the needs, social networks based trust has attracted numerous researchers. We discuss this field in three categories: (1) **Reputation-based**, to infer the trustworthiness of an entity by considering its reputation in social networks, e.g. [32] [20] [38]; (2) **trust relationships based**, to infer indirect trust through

trusted friends, e.g. [11][7] [19]; (3) **vulnerability analysis based**, to evaluate trust indirectly by evaluating how vulnerable is the trust relation network which the trust relies on, e.g. [36] and Advogato [22].

Even though tightly related, trust and reputation are essentially different concepts. The reputation of an entity is the aggregated opinion that a community of people have about how good this entity is. Those people may give their opinions about that entity just based on a single encounter, and may not trust that entity at all; whereas trust is between two entities. Trust means that trustor believes his expectation on trustee to be fulfilled and he is willing to take the risk for that belief.

The rationale for reputation based trust computing is that an agent having high reputation in a domain usually is trustworthy in that domain. So, a reputation metric is frequently used as a substitute of a trust metric. Classical reputation systems (e.g. those used in eBay and Amazon) have been developed in e-commerce[5], which have a central trusted authority to aggregate opinions of users/partners. Some major limitations exist. A rating is usually given by a “stranger” who knows little about the evaluated subject and is limited to just a single interaction; users of reputation metrics don’t know the raters; unfair ratings exist [39]; a very large number of transactions are needed for statistical significance.

From network perspective, Kleinberg [21] proposed a profound model using eigenvector to discover “authorities” and “hubs” in a network. This work has wide influence in succeeded research; Page et al [32] adopted this thought in well known PageRank algorithm (used by Google) to calculate the reputation of a webpage; in the same vein, Kamvar et al [20] developed EigenTrust algorithm using eigenvector to calculate the global trust (actually reputation) from local trust in a P2P network.

Trust relationships based trust computing models infer or calculate indirect trust by using direct trust relationships in social networks. Most of them have two basic trust aggregation operators (or functions): sequence and parallel aggregation. The logic basis of sequence aggregation is transitivity and/or transferability of trust. Generally speaking, if agents share trust data with their trusted friends, within a specific context, *trust in belief* (trust in what trustee believes) is transitive, *trust in performance* (trust in what trustee performs) is not, but can propagates through *trust in belief*[15]. What is the basis for parallel aggregation still remains unclear. Parallel aggregation is an opinion aggregation problem. Ding et al [7] used entropy based aggregation. Many quantitative trust models use various weighted average, appearing as heuristics designed from intuition, for example, the more a trusted friend is, the more the weight of this friend’s opinion is in the aggregation.

Our model is a trust relationships based trust model. We believe, formal semantics is important, because without strictly defined semantics, the meanings of trust and trust degree are vague, the conditions and contexts to apply trust are not well defined, and the implication of trust are unclear, so that trust may be misused; for probability interpretation of trust degrees, it is critical to explicitly define the sample space of that probability. Different from other models in this category, our model has explicitly and formally defined semantics of trust; based on the formal semantics and probability theory, we derive sequence and parallel aggregation operators, rather than define them as heuristics. In this way,

our model is based on a solid formalism foundation.

3. MOTIVATING SCENARIOS

The technology and standards of digital signature and certification with PKI provide a fundamental and secure approach to authentication. An authenticated identity plus a set of authenticated credentials (assertions that the entity has a set of attributes) may lead to authorizing the authenticated entity access to controlled resources such as information, services, or goods, subject to predefined authorization policies. An authorization policy usually is based on either the rights of the authenticated entity or a trust relationship from the resource controller to that entity.

We illustrate the underlying concepts of our research in the following scenarios.

In Chicago, Alice, a physician, needs helps for treating the disease of a patient; in the electronic medical messaging system connecting to her clinic, she creates a message to ask for help, which is sent to a set of doctors with an attribute-based messaging system; soon, Alice receives a message with digital signature from Dan, a specialist in epidemiology in Philadelphia; the message says that Dan could help and he needs further information about the patient; but Alice does not know Dan; can Alice trust Dan regarding Dan’s professional performance and regarding whether Dan follows the terms of use of the privacy data she provides?

First of all, Alice needs to ensure that the message is sent by the person as claimed. To authenticate Dan’s identity, Alice needs to validate Dan’s public key. This is accomplished by discovering a certificate chain from an Alice’s trusted certificate authority (CA) (called trust anchor in PKI literatures) to the CA who issues Dan’s public key, and then by validating each public key in the certificate chain from the public key of Alice’s trust anchor to Dan’s public key. The certification path construction and validation is a standard PKI function. However, any CA in the chain may make mistakes in its certificate such as issuing the certificate to a wrong entity as [10], key compromised for its limited “cryptographic lifetime” or “theft lifetime” [8] before the expired date stated in the certificate, failure in maintaining CRL (Certification Revocation Lists), and so forth. With the growth of the length of a certificate chain, the risk gets higher and higher. How large is this the risk and can it be quantified? If the risk in each certificate chain is quantified, how can this quantified risk be used for certification path construction?

However, the authentication of Dan’s identity is not sufficient for Alice to trust Dan as an expert in epidemiology. This chain of public key certificates is easily misunderstood as a chain of references to Dan’s knowledge of epidemiology, especially when PGP is used for public key certification. Actually, Dan’s public key certificate consists only of (1) Dan’s public key; (2) Dan’s ID information; (3) certification information such as expiration date; (4) a digital signatures signed by a certification authority who verified that the signed public key belongs to Dan, and maintains the validity of this public key. The relationship in a public key certificate is only limited to trust in public key validation.

These scenarios reveal and motivate us to consider a number of relevant issues:

- There are many risks associated with trust in pub-

lic key certificate chains and more general credential chains. These trust relationships are largely dependent on the organizational or human behaviors, which is beyond PKI technology;

- Trust is inherently uncertain, being dependent on behavior of organizations or humans. To support better authentication authorization decisions, there is a need to quantify the risks associated with trust in credential chains;
- The quantified risk associated with trust may be useful in certification path construction and selection;
- Trust is subject to a specific context. Alice may trust a CA regarding issuing and maintaining Dan’s public key, but not regarding whether Dan is a specialist in epidemiology;
- Trust is uncertain. An interesting question here is how to measure the uncertainty in trust in an open environment in which each entity may subjectively give his own trust?
- There may be multiple trust paths between Alice and the CA issuing Dan’s public key. The connection between them may be even more complex as a network. How should the risk associated with a trust network be evaluated?

4. SEMANTICS OF TRUST

4.1 Concept of Trust

Trust is a complex social phenomenon. The concepts developed in social sciences provide an important foundation for trust formalization. A large body of research has contributed to the conceptualization of trust [24] [28][1].

In this paper, we use the following definition of trust [16][15]: *Trust is the psychological state comprising (1) **expectancy**: the trustor expects a specific behavior of the trustee such as providing valid information or effectively performing cooperative actions; (2) **belief**: the trustor believes that the expected behavior occurs, based on the evidence of the trustee’s competence and goodwill; (3) **willingness to be vulnerable**: the trustor is willing to be vulnerable to that belief in a specific context, where the specific behavior of the trustee is expected.*

According to the types of the expectancy in trust, there are two types of trust: *trust in performance* and *trust in belief*. The former is the trust in what trustee performs; the later is the trust in what trustee believes. These two types of trust play important roles in our trust modeling.

4.2 Semantics of Trust and Distrust

Based on the formal semantics of trust defined in the [15], we give a simpler version of the semantics of trust in First Order Logic as follows.

Definition (trust in performance): That trustor d trusts trustee e regarding e ’s performance (represented by x) in context k means that if information x is made by entity e , then entity d believes x in context k .

$$trust_p(d, e, x, k) \equiv madeBy(x, e, k) \supset believe(d, k \dot{\supset} x) \quad (1)$$

In the above definition, information x is a reified proposition¹ representing either an assertion made by e or a commitment made by e to perform (or not to perform) an action; context k is a reified proposition representing the conjunction of a set of “propositions” to characterize a context. A dot over a logical operator is a function to mimic that logical operator, e.g. $\dot{\supset}$ is a function to mimic logical implication.

Definition (trust in belief): That trustor d trusts trustee e regarding e ’s belief (represented by x) in context k means that if information x is believed by entity e , then entity d believes x in context k .

$$trust_b(d, e, x, k) \equiv believe(e, k \dot{\supset} x) \supset believe(d, k \dot{\supset} x) \quad (2)$$

In order to represent uncertainty in trust, the concept of distrust needs to be introduced.

In Marsh’s review on the concepts of trust, untrust, distrust, and mistrust [26], distrust is regarded as the negative form of trust; untrust is a status where the degree of confidence is not enough to trust; mistrust is misplaced trust.

More specifically, in our discussion, *distrust* means trustor d believes the expectancy not to be true, in other words, for *distrust in performance*, the trustor believes that the expected information created by trustee e is false, the expected performance of e does not come true, or unexpected behavior comes true; for *distrust in belief*, the trustor believes what trustee believes is false. Formally,

$$distrust_p(d, e, x, k) \equiv madeBy(x, e, k) \supset believe(d, k \dot{\supset} \neg x) \quad (3)$$

$$distrust_b(d, e, x, k) \equiv believe(e, k \dot{\supset} x) \supset believe(d, k \dot{\supset} \neg x) \quad (4)$$

Here, corresponding to the formal notation of that entity a believes a proposition x , $believe(a, x)$, the formal notation of disbelief is represented by $believe(a, \neg x)$.

This view of distrust is grounded by the theory of Luhmann [24](chapter 10). The literature addresses that trust and distrust are qualitatively different but functionally equivalent; both of them are based on familiarity; both of them reduce social complexity. In other words, both trust and distrust correspond to certainty, but in different directions.

Finally, we define the semantics of a more general form of trust relationships with a specific context but without a specific expectancy.

$$\begin{aligned} trust_b(d, e, k) &\equiv (\forall x, trust_b(d, e, x, k)); \\ trust_p(d, e, k) &\equiv (\forall x, trust_p(d, e, x, k)). \end{aligned} \quad (5)$$

The straightforward meaning of them is that d trust e regarding e ’s every performance or belief in context k . This more general form of trust relationships is more often used in the real world.

Similarly, we define

$$\begin{aligned} distrust_b(d, e, k) &\equiv (\forall x, distrust_b(d, e, x, k)); \\ distrust_p(d, e, k) &\equiv (\forall x, distrust_p(d, e, x, k)). \end{aligned} \quad (6)$$

The general form of distrust is not often in the real world, but logically it is the extreme end of distrust, which will be

¹A reified proposition is a relation representing a “proposition” but it is data rather than a proposition in the representation language.

used in uncertain trust model in the next section.

4.3 Trust Reasoning

The above semantics of trust can be used for trust reasoning. Generally, trust is placed on an entity (or agent), which behaves autonomously. In other words, the behaviors of the trusted entity is out of control of the trustor. On the other hand, belief is placed on information (or more exactly, a proposition). The purposes of formalizing trust are to accurately define what trust means when we use trust, and to use the semantics of trust to infer whether the information created by the trusted entity or the information representing an expected behavior from the trusted entity to be believed to be true.

In the following, we briefly introduce the logical rules for trust reasoning based on the formal semantics of trust.

By applying the formal semantics of trust in performance as well as modus ponens, we have

Rule 1:

$$madeBy(x, e, k) \wedge trust_p(d, e, x, k) \supset believe(d, k \dot{\supset} x) \quad (7)$$

Similarly, for trust in belief, we have:

Rule 2:

$$believe(e, k \dot{\supset} x) \wedge trust_b(d, e, x, k) \supset believe(d, k \dot{\supset} x) \quad (8)$$

By *trust in belief*, *trust in performance* can propagate in a network; in other words, for a given context k , if entity a trusts b on b 's belief in other entities' performance, b trusts c in c 's performance, then a indirectly trusts c regarding c 's performance.

However, when entity a trusts b on belief in a context, and b trusts c on performance in a *different* context, from these two trust relationships, we cannot derive that a trusts c .

The rules for trust propagation are given as follows.

Rule 3:

$$trust_b(a, b, x, k) \wedge trust_p(b, c, x, k) \supset trust_p(a, c, x, k) \quad (9)$$

$$trust_b(a, b, x, k) \wedge trust_b(b, c, x, k) \supset trust_b(a, c, x, k) \quad (10)$$

This rule requires the expectancy (x) of trust to be the same in the trust from a to b and the trust from b to c . Actually, this rule can be extended to a more general form without a specific expectancy, as follows.

Rule 4:

$$trust_b(a, b, k) \wedge trust_p(b, c, k) \supset trust_p(a, c, k) \quad (11)$$

$$trust_b(a, b, k) \wedge trust_b(b, c, k) \supset trust_b(a, c, k) \quad (12)$$

The proof of these trust propagation rules can be found in [15] and is omitted here.

5. MEASUREMENT OF UNCERTAIN TRUST

Because trust is placed on another organization, another person, or a group of persons, the features of human behaviors make trust inherently uncertain. In this section, we discuss the measurement and representation of uncertain trust. We use probability to measure uncertainty in trust, so that each entity in a distributed computing environment could give his degrees representing uncertainty in trust, based on a common understanding regarding what the numbers mean.

5.1 Formal Definition of Trust Degree

Based on the formal semantics of trust presented in the previous section, as well as the connections of probability and conditionals [13] studied in philosophical logic, the degree of trust is defined as follows.

$$td^p(d, e, x, k) = pr(believe(d, x) | madeBy(x, e, k) \wedge beTrue(k)) \quad (13)$$

for trust in performance;

$$td^b(d, e, x, k) = pr(believe(d, x) | believe(e, x) \wedge beTrue(k)) \quad (14)$$

for trust in belief.

When the type of trust is not concerned, we omit the superscript p/b .

Similar to trust degree, based on the formal semantics of distrust, the degree of distrust is defined as:

$$dtd^p(d, e, x, k) = pr(believe(d, \dot{\supset}x) | madeBy(x, e, k) \wedge beTrue(k)) \quad (15)$$

for distrust in performance;

$$dtd^b(d, e, x, k) = pr(believe(d, \dot{\supset}x) | believe(e, x) \wedge beTrue(k)) \quad (16)$$

for distrust in belief.

The sample space of the probability representing trust degree could be any event set that contains the events in which the conditions are true. The minimal sample space is exactly the set of events in which the conditions in the conditional probability are true.

5.2 Measurement of Trust Degree

The previous subsection provides formal definitions of trust /distrust degrees in probability theory. This subsection gives a frequency interpretation to the probabilities defining trust /distrust degrees. The formal definitions and the frequency interpretation provide a formal interpretation about the semantics of the numbers representing trust /distrust degrees, and puts the calculus of trust in a firm theoretic basis. The latter point is important—in practice, this frequency interpretation of trust /distrust degree can be used as a practical method to calculate trust /distrust degrees, by using the data accumulated in the interaction between a trustor and a trustee. However, when the data of interactions are not available or not used, a trust /distrust degree may be given as a subjective probability but with the assurance that the calculus itself is nevertheless correct.

In the following, we describe measurement of trust /distrust degrees and a frequency interpretation of probability.

Trust can be divided into two categories: (1) *direct trust*, the trust coming from direct interaction between two parties, and (2) *indirect trust*, the trust derived from a social network. The derivation of indirect trust will be discussed in the next two sections; we only need to discuss how to count direct trust.

The degree of trust is measured with the frequency rate of trustor's positive experience among all encounters with the trustee. That is,

$$td(d, e, x, k) = n/m, \quad (17)$$

where, m is the total number of encounters regarding an instanced expectancy x , and n is the number of trustor's

positive experience. For example, x is an assertion about the authentication of a specific customer, John, who signs on to request a service; d is the service provider; e is the identity provider who makes authentication assertion x ; k is a context such as “sign in for online shopping”; m is the total times of which e informed d about the authentication of John’s Id in d ’s historic data set; n is the number of correct authentication about John; this rate of n/m reflects the probability by which e makes correct authentication about John’s Id.

similarly, we have

$$dtd(d, e, x, k) = l/m, \quad (18)$$

where l is the number of trustor’s negative experience. $n + l$ is not necessarily equivalent to m , if some encounters are hard to say being positive or negative.

For the degree of general trust without a specific expectancy,

$$\begin{aligned} td(d, e, k) &= n'/m', \\ dtd(d, e, k) &= l'/m' \end{aligned} \quad (19)$$

where, m' is the number of all encounters between trustor and trustee regarding all instanced expectancy ($\forall x$). n' is the number of positive experience in those encounters. l' is the number of negative experience in those encounters. In the earlier example, m becomes the total times of which e informed d about the authentication of the identity of any signed custom in d ’s historic data set; n becomes the number of correct authentication.

In practice, people use specific information for a specific problem solving, but when the specific information is not available, the general inform may be applied. So, for the case of lack of information about a specific expectancy x , $td(d, e, k)$ may be used as an estimated value of $td(d, e, x, k)$.

Similar to uncertain belief and uncertain trust, a trustor may evaluate each encounter as positive (or satisfied, succeed) to a certain extent, as negative (or unsatisfied, failed) to a certain extent, and as undecidable (or hard to say positive or negative) in a certain degree. In such case, trust /distrust degrees can be refined as:

$$td(d, e, x, k) = \frac{\sum_{i=1}^m e_p(i)}{m}, \quad (20)$$

where m is the same as defined earlier;

$$e_p(i) \in [0, 1]$$

represents the degree of encounter i being positive, $e_p(i) = 1$ represents completely positive, and $e_p(i) = 0$ represents completely not positive.

Similarly, for distrust degree,

$$dtd(d, e, x, k) = \frac{\sum_{i=1}^m e_n(i)}{m}, \quad (21)$$

where,

$$e_n(i) \in [0, 1],$$

which is the degree of encounter i being negative, and

$$e_p(i) + e_n(i) \leq 1. \quad (22)$$

The difference,

$$1 - e_p(i) - e_n(i) \quad (23)$$

represents the degree of uncertainty in trustor’s evaluation of encounter i due to lack of sufficient information for judgment.

5.3 Notation of Uncertain Trust Relationships

A trust relationship can be represented by the degree of trust and the degree of distrust.

The sum of degrees of trust and distrust actually represents the degree of certainty, denoted as cd , e.g.

$$cd(d, e, x, k) = td(d, e, x, k) + dtd(d, e, x, k). \quad (24)$$

The degree of uncertainty, denoted as ud , is defined as

$$\begin{aligned} ud(d, e, x, k) &= 1 - cd(d, e, x, k) \\ &= 1 - td(d, e, x, k) - dtd(d, e, x, k). \end{aligned} \quad (25)$$

This uncertainty comes from the unfamiliarity of trustor to trustee, or from a trustor’s lack of sufficient information to evaluate trust of the trustee.

We have $td + dtd + ud = 1$.

When $ud = 0$, $td + dtd = 1$, which corresponds to the most certain situation in which the trustor is sufficiently familiar with the trustee, so the trustor can surely make decision to either trust or distrust the trustee.

When $0 < ud < 1$, $td + dtd < 1$, which corresponds to a typical uncertain situation in which the trustor is not sufficiently familiar with the trustee, so there is uncertainty regarding trust decision.

When $ud = 1$, $td + dtd = 0$, which corresponds to the most uncertain situation in which the trustor is completely not know about the trustee, so the trustor cannot give the degrees of trust or distrust at all. This is the typical case of “untrust”

Another interesting case is $td = dtd = 0.5$, which is the most uncertain situation when $ud = 0$.

A situation easy to be confused is when $td = 0$. Some people may think it means untrust; some may think it means distrust; some others may think it’s not sure which case is. If a trust relationship is defined sole by td , it is hard to distinguish between these two possibilities. In our notation, when $td = 0$, $ud + dtd = 1$. So, depending on what ud and dtd are, there is a probability distribution over distrust and untrust.

Therefore, a trust relationship can be formally represented as

$$tr(d, e, x, k) = \langle td(d, e, x, k), dtd(d, e, x, k) \rangle, \quad (26)$$

or

$$tr(d, e, x, k) = \langle td(d, e, x, k), dtd(d, e, x, k), ud(d, e, x, k) \rangle, \quad (27)$$

when the value of ud need to appear explicitly.

For a general trust relationship without a specific expectancy, we have

$$tr(d, e, k) = \langle td(d, e, x, k), dtd(d, e, k) \rangle. \quad (28)$$

This formal representation of trust relationships has strictly defined semantics, so that it can help to avoid mistakes as we develop a calculus for trust.

6. TRUST CALCULATION

In the following, we discuss how to calculate the degree of trust when trust propagates in trust networks. Basically, there are two types of trust aggregations: sequence aggregation and parallel aggregation. Sequence aggregation describes aggregation of trust degrees along a trust path; parallel aggregation is about how to aggregate trust degrees in several parallel trust paths.

6.1 Sequence Trust Aggregation

The sequence aggregation problem is this: given that a trusts b (on belief) with a probability distribution trust, distrust, and undecidable, b trusts c (either on belief or on performance) with another probability distribution, what is the probability distribution for a to trust c , that is, what are $td(a, c, x, k)$ and $td(a, c, x, k)$?

The following theorem answers this question.

Theorem UT-1: (1) assume that agent a has a trust in belief relationship with b ,

$$tr(a, b, x, k) = \langle td^b(a, b, x, k), dtd^b(a, b, x, k) \rangle, \quad (29)$$

b has trust in performance relationship with c ,

$$tr(b, c, x, k) = \langle td^p(b, c, x, k), dtd^p(b, c, x, k) \rangle, \quad (30)$$

and the belief of a in x is conditionally independent to the provenance of x (or the belief of c in x) given the belief of b in x , then the trust relationship from a to c can be derived as follows:

$$tr(a, c, x, k) = \langle td^p(a, c, x, k), dtd^p(a, c, x, k) \rangle, \quad (31)$$

and

$$td^p(a, c, x, k) = td^b(a, b, x, k) \cdot td^p(b, c, x, k) + dtd^b(a, b, x, k) \cdot dtd^p(b, c, x, k), \quad (32)$$

$$dtd^p(a, c, x, k) = dtd^b(a, b, x, k) \cdot td^p(b, c, x, k) + td^b(a, b, x, k) \cdot dtd^p(b, c, x, k); \quad (33)$$

The certainty degree in this derived trust relationship satisfies

$$cd(a, c, x, k) = cd(a, b, x, k) \cdot cd(b, c, x, k). \quad (34)$$

(2) if b has trust in belief relationship with c , and the conditional independent condition is – the belief of a in x is conditionally independent to the belief of c in x given the belief of b in x , then the trust relationship from a to c can also be derived, but the derived trust is trust in belief.

<end of theorem>

For reasons of space, the proof of this theorem is omitted here but can be found in [17].

The above assumed conditional independent condition is similar to the assumption in belief networks and Markov chains, which assumes an event is only directly dependent on its parents.

For general trust relationships without a specific expectancy, the above theorem also true by removing all expectancy x , and revising the conditional independent condition as follows: the belief of a in any x is conditionally independent to the provenance of any x (or the belief of c in any x) given the belief of b in any x .

This sequence aggregation operator has some interesting properties.

Property 1: with the growth of the length of a trust path, the certainty degree of the aggregated trust multiplicatively decreases.

For simplicity, we use subscript i, j represents that a trust relationship is from entity i to entity j .

Assume the length of a trust path is n . By theorem UT-1, we have

$$cd_{1,n} = cd_{1,2} \cdot cd_{2,3} \cdot \dots \cdot cd_{n-1,n}, \quad (35)$$

Therefore, we have property-1.

This property is coincident with people’s intuition regarding trust decreasing quickly in propagation along a trust path.

Property 2: sequence aggregation is associative.

By this property, the outcome of the sequence aggregation is independent to the order of sequence aggregation for each pair of trust relationships in a trust path. This property is important when applying sequence aggregation in the algorithm for trust aggregation in a network.

Property 3: A trust relationship with $ud = 1$ is a “zero” element in trust aggregation.

This property says that if the trust of a in b or the trust of b in c is untrust with $ud = 1$, the derived trust of a in c is also “untrust” with $ud = 1$. In other words, in a trust network, a trust relationship with $ud = 1$ is the same as there is no trust relation between the two entities.

The implication of this property is obvious. In a trust network, if there is only one trust path from a to c , then any “untrust” in the path will make the trust path “broken” which is equivalent to the case that there is no trust path from a to c . As a result, a will “untrust” c .

This property reveals that trust evaluation will not change by adding or cutting off a trust relationship with $td = dtd = 0$. Therefore, cutting off all trust relationships with td and dtd near 0 will effectively reduce the complexity of a trust network and the associated computation complexity.

Sequence aggregation is a basis for trust aggregation in a trust network. Sequence aggregation also can be applied independently. For example, in identity management, a credential chain is a trust path, and sequence aggregation can be used for analyzing trust related risk in a credential chain. In PGP, a sequence of public key introducers forms a trust path, and sequence aggregation can be used to calculate a numeric value of the overall trust along that trust path.

A key issue now is aggregation of trust relationships (opinions) in parallel trust paths, which is difficult due to the lack of the commonly recognized logic to synthesize different opinions. In the following, we discuss how to make parallel trust aggregation first, then we discuss how to make trust evaluation in a trust network by using sequence aggregation and parallel aggregation.

6.2 Parallel Trust Aggregation

In general, as shown in figure 1, parallel trust aggregation needs to answer the following question: given that entity a directly or indirectly trusts (in belief) $b_1, \dots, b_n, b_1, \dots, b_n$ trust c (either in belief or in performance), and a may also directly trusts c , what is the aggregated trust from a to c ?

We assume that in a trust network each direct trust relationship (described by trust degree and distrust degree) and the number of samples used to determine that trust relationship are given.

For simplicity, we use $td(e_i, e_j)$ ($dtd(e_i, e_j)$) to denote $td(e_i, e_j, x, k)$ ($dtd(e_i, e_j, x, k)$), by omitting x, k , because they are the same²; we use $s(e_i, e_j)$ to denote the number of samples used in assessing $td(e_i, e_j)$ and $dtd(e_i, e_j)$; furthermore, we use a superscript $*$ to denote an aggregated trust relationship, e.g. $td^*(e_i, e_j)$ ($dtd^*(e_i, e_j)$). In addition, we omit superscripts p and b when the type of trust is not

²Actually, for a specific trust evaluation regarding the information x in context k , a specific sub-network with the same x and k is selected from a real world trust network. See detail in subsection 6.3.

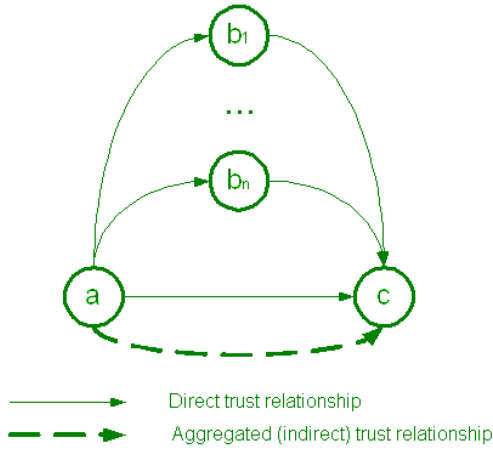


Figure 1: Parallel trust aggregation in multiple trust paths

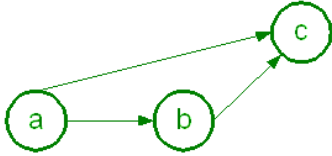


Figure 2: A simple example of parallel trust aggregation

concerned.

We start from the simplest case as shown in figure 2. Here entity a has two trust paths to entity c : the direct trust from a to c , and the indirect trust via b .

We define parallel aggregation based on the interpretation of a trust degree as a frequency rate of successful interaction. From the view of entity a , the total number of encounters with c regarding the information x in context k is the sum of the encounters of both entity a 's direct interaction and indirect interaction via b with c , that is,

$$s^*(a, c) = s(a, c) + s(a, b, c), \quad (36)$$

where,

$$s(a, b, c) = s(b, c); \quad (37)$$

among these encounters, the total number of successful encounters with c is the sum of (1) the successful encounters that a directly interacts with c , which is, by frequency rate definition of trust degree,

$$s(a, c) \cdot td(a, c); \quad (38)$$

and (2) the successful encounters that a indirectly interact via b with c , which is, by sequence trust aggregation,

$$s(a, b, c) \cdot (td^b(a, b) \cdot td(b, c) + dtd^b(a, b) \cdot dtd(b, c)). \quad (39)$$

A natural interpretation is that a reviews each direct interaction that b had with c , and evaluates each certain (i.e., positive or negative) interaction (from a 's point of view) as being the same as that from b 's point of view with probability $td^b(a, b)$, and being the opposite of b 's with probability $dtd^b(a, b)$. In a sense b 's direct interactions with c are being interpreted as a 's interactions with c , and are given the same

weight (after normalization by trust-in-belief measures) as direct interactions that a has with c .

So, the aggregated degree of trust from entity a to c , $td^*(a, c)$, is:

$$\begin{aligned} td^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot td(a, c) \\ &+ (s(a, b, c)/s^*(a, c)) \cdot \\ &(td^b(a, b) \cdot td(b, c) \\ &+ dtd^b(a, b) \cdot dtd(b, c)) \end{aligned} \quad (40)$$

The type of $td^*(a, c)$ depends on the type of $td(b, c)$. For $td^p(b, c)$, $td^*(a, c)$ will be trust in performance; for $td^b(b, c)$, $td^*(a, c)$ will be trust in belief.

Similarly, we have the aggregated degree of distrust as follows.

$$\begin{aligned} dtd^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot dtd(a, c) \\ &+ (s(a, b, c)/s^*(a, c)) \cdot \\ &(td^b(a, b) \cdot td(b, c) \\ &+ dtd^b(a, b) \cdot dtd(b, c)) \end{aligned} \quad (41)$$

For the general case shown in figure 3, the aggregated trust degree can be calculated as

$$\begin{aligned} td^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot td(a, c) \\ &+ \sum_{i=1, \dots, n} (s(a, b_i, c)/s^*(a, c)) \cdot \\ &(td^b(a, b_i) \cdot td(b_i, c) \\ &+ dtd^b(a, b_i) \cdot dtd(b_i, c)), \end{aligned} \quad (42)$$

where

$$s^*(a, c) = s(a, c) + \sum_{i=1, \dots, n} s(a, b_i, c), \quad (43)$$

and

$$s(a, b_i, c) = s(b_i, c); \quad (44)$$

Similarly, the aggregated distrust degree can be calculated as

$$\begin{aligned} dtd^*(a, c) &= (s(a, c)/s^*(a, c)) \cdot dtd(a, c) \\ &+ \sum_{i=1, \dots, n} (s(a, b_i, c)/s^*(a, c)) \cdot \\ &(td^b(a, b_i) \cdot dtd(b_i, c) \\ &+ dtd^b(a, b_i) \cdot td(b_i, c)); \end{aligned} \quad (45)$$

From the frequency definition of trust degree, a parallel aggregation is derived, which appears in the form of weighted average of all parallel trust paths, and the weight of a path is the rate of the number of the samples in this path to the total number of samples.

Generally, the more samples are used in a trust path, the more accurate the trust degree is in that trust path, and the more weight that trust path will have in aggregation. This is also coincident with people's intuition regarding opinion aggregation.

Parallel aggregation is associative, so that the aggregated trust relationship will not change with the order of aggregation. This property, together with the associativity of sequence aggregation, is important to make the calculated result unique in different algorithm implementations which may aggregate trust paths in different order.

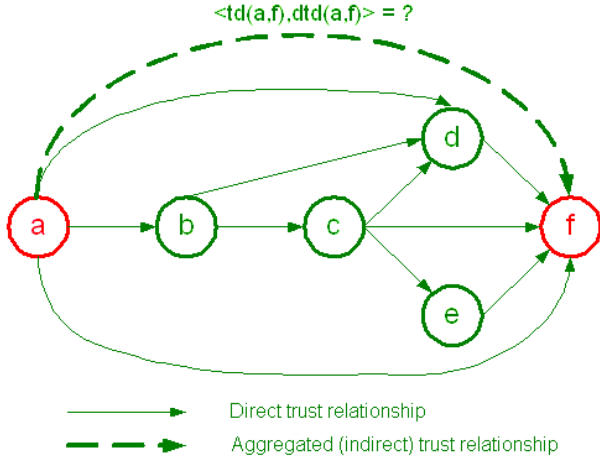


Figure 3: Example: trust aggregation in trust networks – What is the aggregated trust relationship between a and f ?

Finally, it is interesting to interpret this parallel aggregation from the view of trust revision with new information. $td(a, c)$ can be regarded as the initial opinion of entity a regarding trust in entity c , based on a 's direct interaction with c ; later, a learns new information from her/his trusted friends b_1, \dots, b_n regarding their opinion about c , and then a revises her/his opinion as $td^*(a, c)$, by synthesize her/his friends' opinions. If a 's opinion is based on a small number of direct interaction, those friends' opinions may have large influences on her/his revised opinion; on the other hand, if her/his original opinion is based on a big number of samples, her/his friends opinions may have smaller influences.

6.3 Trust Evaluation in a Network

We now show how to use sequence aggregation and parallel aggregation to aggregate trust in a network.

A *trust network* is a subgraph of a social network. A *social network* can be regarded as a directed graph with labeled arcs, where the vertices are entities such individuals and organizations in society, and the arcs are various social relationships, typically acquaintance relationships. In the context of trust, we only concern a special type of subgraphs of social networks, called *trust networks*, in which arcs represent inter-individual (or direct) trust relationships. An arc from vertex a to vertex b represents that a has direct trust relationship with b which is described as $\langle td(a, b, x, k), dtd(a, b, x, k) \rangle$. In general, all direct trust relationships in a social network form a directed graph usually with circles. For the purpose of deriving a new trust relationship by a trust network, trust circle should be eliminated. For this reason, we assume a concerned trust network is a directed acyclic graph (DAG).

More specifically, for a specific trust evaluation from trustor a to trustee z regarding information x in context k , we only consider a sub-network (of a real world trust network) with source node a and sink node z and the arcs in which trust relationships have the same x and k . Because in this specific subset of a trust network, all trust relationships have the same x and k , they are omitted.

From this point of view a trust network with source a and

sink z is a DAG (directed acyclic graph), represented as

$$TN = (E, A); \quad (46)$$

E is the set of entities, and $a, z \in E$; $A \in E \times E$, and each arc (e_i, e_j) ($e_j \neq z$) is labeled by trust (in belief) relationship

$$\langle td^b(e_i, e_j), dtd^b(e_i, e_j) \rangle; \quad (47)$$

each arc to the sink, (e_i, z) , is labeled by trust relationship

$$\langle td^x(e_i, e_j), dtd^x(e_i, e_j) \rangle \quad (48)$$

x is either b or p , that is, all arcs to the sink are either trust in belief relationship or trust in performance relationship.

An algorithm is designed to make trust aggregation in a network, which recursively simplifies a more complex network to a simpler one, by replacing multiple parallel paths into a single arc. Each replacement is made by using sequence or parallel aggregation.

$aggregate(a, z, TN)\{$

if (a, z) *is the only path from a to z in TN, stop;*
else if a has and only has one path to z, then $\{$

use sequence aggregation to aggregate;
remove the last arc in this path to z;
add arc (a, z) labeled by $td^(a, z)$ in TN;*

$\}$ *else if a has multiple disintersected paths to z,*
then $\{$

use parallel aggregation to aggregate all paths from a to z;
remove the last arc in each path to z;
add arc (a, z) labeled by $td^(a, z)$ in TN;*

$\}$ *else* $\{$

calculate $N = neighbors(z)$;
for each $n_i \neq a$ in N do $aggregate(a, n_i, TN)$;
use parallel aggregation to aggregate all paths from a to z;
remove the last arc in each path to z;
add arc (a, z) labeled by $td^(a, z)$ in TN;*

$\}$

$\}$

The example shown in figures 3 to 7 demonstrates the trust aggregation process in a trust network.

In figure 3, the set of f 's neighbors is $\{a, c, d, e\}$. Apply the algorithm to aggregate trust in each neighbor first.

Figure 4 shows the process to aggregate trust between a and d . Since the subgraph with a as source and d as sink is still a network, apply the algorithm again. The neighbors of d are b and c ; a has direct trust in b , so no aggregation is applied; a has a single path to c , apply sequence aggregation, then remove the last arc (b, c) in the path, and add arc (a, c) (bold dash arc) corresponding to the aggregated trust relationship $\langle td^*(a, c), dtd^*(a, c) \rangle$ by the sequence aggregation. The result is shown in in figure 4 (b); now, return to the aggregation between a and d . a has three parallel paths to d , (a, d) , (a, b, d) and (a, c, d) . Apply parallel aggregation, then remove the last arc in each path, i.e. (a, d) , (b, d) and (c, d) , add a new arc, (a, d) (bold dash arc), which is corresponding to the aggregated trust relationship $\langle td^*(a, d), dtd^*(a, d) \rangle$ by the parallel aggregation. The result is shown in figure 4 (c).

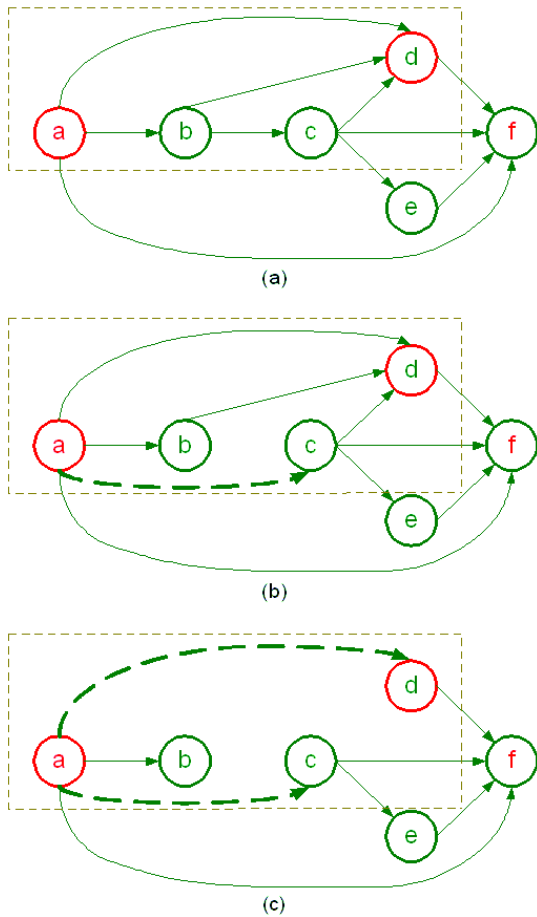


Figure 4: Example: trust aggregation in trust networks – aggregation between a and d

In figure 5, to aggregate trust between a and c , a has an arc to c , so the algorithm do nothing, and return to upper level of the process to aggregate trust between a and f .

Figure 6 shows the process to aggregate trust between a and e . There is a single path (a, c, e) , so apply sequence aggregation, then remove arc (c, e) and add arc (a, e) (bold dash arc) corresponding to the aggregated trust relationship by the sequence aggregation.

Returning to the process of aggregating trust between a and f , as shown in figure 7(a), there are four parallel trust paths, (a, f) , (a, d, f) , (a, c, f) and (a, e, f) , as shown in (a). Apply parallel aggregation, remove the last arc of each path, i.e. (a, f) , (d, f) , (c, f) and (e, f) , and add the new arc (a, f) (red bold dash arc) corresponding to the aggregated trust relationship by the parallel aggregation. Thus, we obtain the aggregated trust between a and f , as shown in 7(b).

7. TRUST QUANTIFICATION IN CERTIFICATE CHAINS

As discussed in section 1, a number of PKI trust models have been proposed [33] [23] [30] [37]. However, the explicit and accurate representation of trust relationships and the quantification of risks associated with trust in certification paths are missing.

In PKI, a certification path actually corresponds to a chain

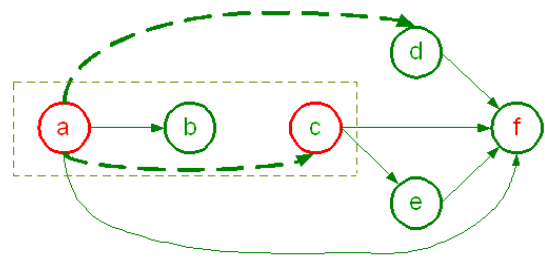


Figure 5: Example: trust aggregation in trust networks – no aggregation needed between a and c

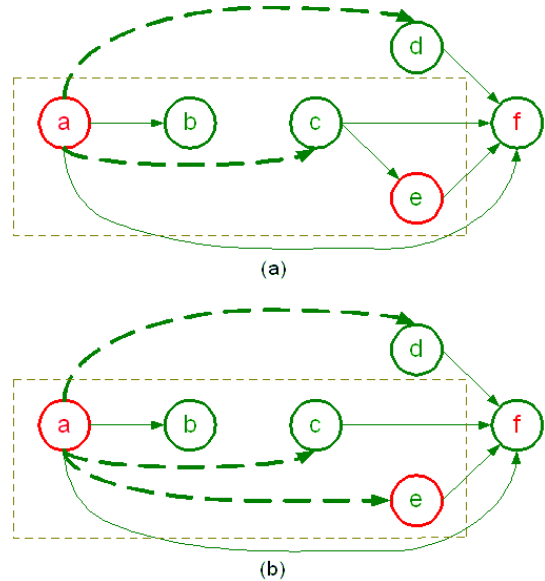


Figure 6: Example: trust aggregation in trust networks – Sequence aggregation between a and e

of trust relationships. What do these trust relationships exactly mean? On what things does each entity trust another in a credential chain? What is the specific context of each trust? In order to avoid misuse trust in PKI, we need to answer these questions and to explicitly and accurately represent trust in certificate chains.

In a typical public key validation model, a single certification path with shortest length is discovered and validated. An implicit risk evaluation criterion here is that a longer certification path has higher risk. This criterion actually assumes each certificate has the same level of risk. However, different certificates have different levels of risk, as they are produced by different organizations, with different identity standards. In order to make better decisions on authentication and authorization, it is important to quantify the risk associated with trust in certificate chains. When quantified evaluation of risk is introduced to certification paths, some very interesting issues emerge. There are multiple available certification paths, but which certification path should be chosen? The shortest path? or the one most trusted? Should more than one certification paths be considered? Should all certificate paths be considered? In this section, we explore the answers to these questions.

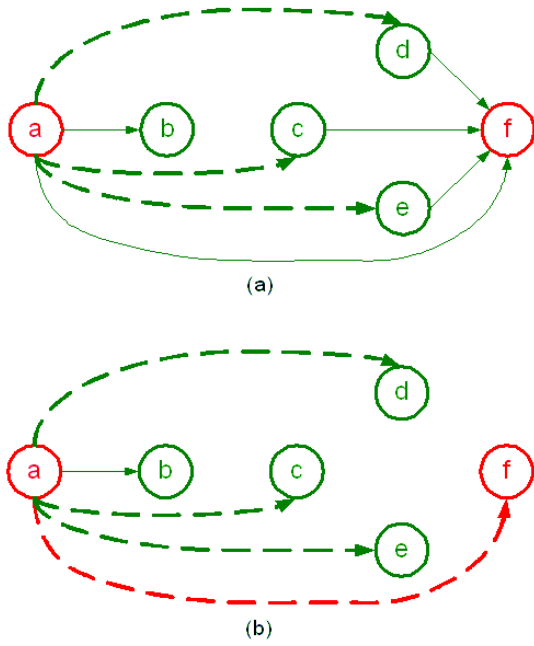


Figure 7: Example: trust aggregation in trust networks – parallel aggregation between a and f

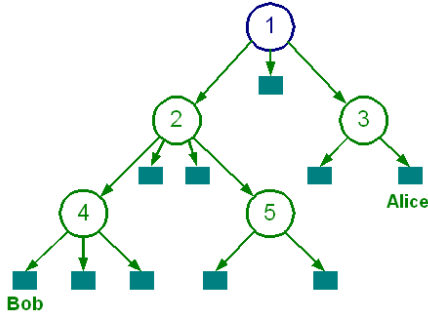


Figure 8: An example of hierarchical PKI (cited from Burr (1998))

In the following, we discuss the semantics of trust in PKI, the formal representation of trust relationships in PKI, and the quantified evaluation of risk associated with trust in certificate chains, by using our calculus of trust. There are different types of PKI architectures [3] [34][14] such as single-CA structure, hierarchical structure, mesh structure, bridge structure, and hybrid structure. We mainly discuss two representative types: hierarchical and mesh.

7.1 Trust in Hierarchical Structure PKI

An example of hierarchical PKI structure is shown in figure 8.

Alice needs to validate Bob’s public key. CA3 is Alice’s trust anchor, the CA Alice trusts; CA1 is root CA that everyone including Alice knows its public key; CA4 issues Bob’s public key certificate. In this case, the certificate chain will be CA1 - CA2 - CA4.

The semantics of the trust relationships in this certificate chain are as follows. The trust from CA2 to CA4 is that CA2

trusts CA4 regarding the validity of the certificates created and maintained by CA4. The trust from CA1 to CA2 is that CA1 trusts CA2 regarding CA2’s performance in the digital certification business, including (1) issuing and maintaining certificates to CA2’s clients, (2) auditing CA2’s subordinate CAs. The latter implies that CA1 trusts CA2 regarding what CA2 believes about the validity of the certificates created and maintained by CA2’s subordinate CAs. Similarly, the trust from Alice to CA1 is that Alice trusts what CA1 believes about the validity of the certificates created and maintained by CA1’s subordinate CAs and all descendants.

In the terminology of our formal trust model, CA2 trusts CA1 on performance in the context of “issuing and maintaining certificates”; CA1 trusts CA2 on belief in the context of “issuing and maintaining certificates”; Alice trusts CA1 in the same context. Depending the application, the context of trust could be more specific issues such as “accurate validation of key holder’s identity” and “good maintenance of CRL”.

Using our formal notation, the above trust relationships can be formally represented and calculated. The expectancy of Alice on Bob is that “public key pk(Bob) is Bob’s”. So, let x be this proposition. In this example, context k is set as “issuing and maintaining certificates”. Assume each of the above trust relationships has a different level of trust. They are formally represented as follows.

$$\begin{aligned}
 tr^b(Alice, CA1, k) &= \langle td^b(Alice, CA1, k), dtd^b(Alice, CA1, k) \rangle \\
 td^b(Alice, CA1, k) &= 0.98; \\
 dtd^b(Alice, CA1, k) &= 0.01; \\
 ud^b(Alice, CA1, k) &= 0.01;
 \end{aligned} \tag{49}$$

$$\begin{aligned}
 tr^b(CA1, CA2, k) &= \langle td^b(CA1, CA2, k), dtd^b(CA1, CA2, k) \rangle \\
 td^b(CA1, CA2, k) &= 0.92; \\
 dtd^b(CA1, CA2, k) &= 0.02; \\
 ud^b(CA1, CA2, k) &= 0.06;
 \end{aligned} \tag{50}$$

$$\begin{aligned}
 tr^p(CA2, CA4, k) &= \langle td^p(CA2, CA4, k), dtd^p(CA2, CA4, k) \rangle \\
 td^p(CA2, CA4, k) &= 0.96; \\
 dtd^p(CA2, CA4, k) &= 0.01; \\
 ud^p(CA2, CA4, k) &= 0.03;
 \end{aligned} \tag{51}$$

For hierarchical PKI the certification path is unique so we can directly apply sequence trust aggregation to evaluate the overall risk in the certificate path.

Using sequence aggregation, the aggregated trust relationship from CA1 to CA4 is calculated as

$$\begin{aligned}
 tr^p(CA1, CA4, k) &= \langle td^p(CA1, CA4, k), dtd^p(CA1, CA4, k) \rangle \\
 td^p(CA1, CA4, k) &= 0.883; \\
 dtd^p(CA1, CA4, k) &= 0.028; \\
 ud^p(CA1, CA4, k) &= 0.09;
 \end{aligned} \tag{52}$$

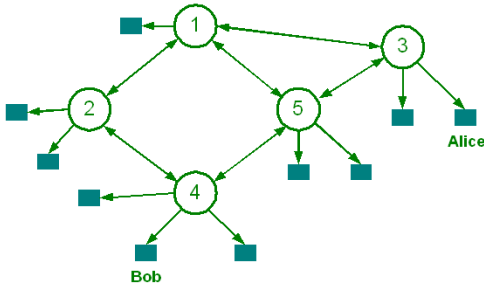


Figure 9: An example of mesh PKI, cited from Burr (1998)

The aggregated trust relationship from Alice to CA4 is calculated as

$$\begin{aligned}
 tr^P(Alice, CA4, k) &= \langle td^P(Alice, CA4, k), dtd^P(Alice, CA4, k) \rangle \\
 td^P(Alice, CA4, k) &= 0.866; \\
 dtd^P(Alice, CA4, k) &= 0.037; \\
 ud^P(Alice, CA4, k) &= 0.097;
 \end{aligned} \tag{53}$$

The single-CA PKI is a special case of hierarchical structure, with only a root CA. The semantics of trust, formal representation, and calculation are the same as in hierarchical PKI.

7.2 Trust in Mesh Structure PKI

An example of mesh PKI is illustrated in figure 9. In this case, CA3 is Alice's trust anchor, i.e. an CA Alice trusts, and by this CA, Alice finds a certificate chain to CA4, the issuer of Bob's public key certificate.

For different structures, while the certification path (certificate chain) construction methods [9][40] differ, from the view of a relying party (verifier or recipient), the trust relationships in a certificate chain have the same semantics.

The semantics of the trust relationships in this example are as follows.

In the following pairs of CAs, CA1 and CA2, CA2 and CA4, CA1 and CA5, CA4 and CA5, CA1 and CA3, as well as CA3 and CA5, each pair of CAs trust each other regarding the validity of the certificates created and maintained by the other; and each pair of CAs trust each other regarding what the other believes about the validity of the certificates created and maintained by a third CA. In the terminology of our formal trust model, each pair of CAs trust each other on both performance and belief in the context of "issuing and maintaining certificates"; Alice trusts CA3 on both performance and belief in the context of "issuing and maintaining certificates".

The trust relationships in a bridge PKI are similar to the ones in mesh PKI. The difference is that bridge CAs play the role of gateway and issue certificates only to CAs, so all trust relationships to bridge CAs are *trust in belief* type. A hybrid PKI consists of many CA groups of different types, so some gateway CAs have trust relationships cross groups; and some others has trust relationships within their groups.

The formal representation of the above trust relationships are similar to the ones given in hierarchical structure. In the following discussion, we give only that part of them needed for the trust calculation examples.

As discussed earlier, when introducing quantified values of trust in certificate chains, we need to answer a number of interesting questions. We discuss those questions in two representative cases.

7.2.1 Using One-path certification

By a traditional certification path construction method, the shortest path, CA3->CA5->CA4, will be used to validate Bob's public key. Assume that the trust relationships are as follows.

Alice highly trust her trust anchor, CA3.

$$\begin{aligned}
 tr^b(Alice, CA3, k) &= \langle td^b(Alice, CA3, k), dtd^b(Alice, CA3, k) \rangle \\
 td^b(Alice, CA3, k) &= 0.99; \\
 dtd^b(Alice, CA3, k) &= 0.0; \\
 ud^b(Alice, CA3, k) &= 0.01;
 \end{aligned} \tag{54}$$

Trust from CA3 to CA5 is not high, even somewhat negative.

$$\begin{aligned}
 tr^b(CA3, CA5, k) &= \langle td^b(CA3, CA5, k), dtd^b(CA3, CA5, k) \rangle \\
 td^b(CA3, CA5, k) &= 0.65; \\
 dtd^b(CA3, CA5, k) &= 0.25; \\
 ud^b(CA3, CA5, k) &= 0.1;
 \end{aligned} \tag{55}$$

The trust from CA5 to CA4 is fairly uncertain.

$$\begin{aligned}
 tr^P(CA5, CA4, k) &= \langle td^P(CA5, CA4, k), dtd^P(CA5, CA4, k) \rangle \\
 td^P(CA5, CA4, k) &= 0.75; \\
 dtd^P(CA5, CA4, k) &= 0.0; \\
 ud^P(CA5, CA4, k) &= 0.25;
 \end{aligned} \tag{56}$$

By sequence aggregation, the derived trust from CA3 to CA4 is

$$\begin{aligned}
 tr^P(CA3, CA4, k) &= \langle td^P(CA3, CA4, k), dtd^P(CA3, CA4, k) \rangle \\
 td^P(CA3, CA4, k) &= 0.488; \\
 dtd^P(CA3, CA4, k) &= 0.188; \\
 ud^P(CA3, CA4, k) &= 0.324;
 \end{aligned} \tag{57}$$

the derived trust from Alice to CA4 is

$$\begin{aligned}
 tr^P(Alice, CA4, k) &= \langle td^P(Alice, CA4, k), dtd^P(Alice, CA4, k) \rangle \\
 td^P(Alice, CA4, k) &= 0.483; \\
 dtd^P(Alice, CA4, k) &= 0.186; \\
 ud^P(Alice, CA4, k) &= 0.331,
 \end{aligned} \tag{58}$$

which shows a weak trust relationship from Alice to CA4, so even though certification is verified successfully along the certificate chain, Alice may still not trust the validity of Bob's public key.

However, a longer path, CA3 -> CA1 -> CA2 -> CA4, with a higher level of trust, may make the derived trust from Alice to CA4 in a acceptable level.

Assume trust relationship from CA3 to CA1 being

$$\begin{aligned}
& tr^b(CA3, CA1, k) \\
& = \langle td^b(CA3, CA1, k), dtd^b(CA3, CA1, k) \rangle \\
& td^b(CA3, CA1, k) = 0.98; \\
& dtd^b(CA3, CA1, k) = 0.01; \\
& ud^b(CA3, CA1, k) = 0.01;
\end{aligned} \tag{59}$$

other trust relationships are as assumed earlier.

By sequence aggregation along this longer path, the derived trust relationship from CA3 to CA4 is:

$$\begin{aligned}
& tr^p(CA3, CA4, k) \\
& = \langle td^p(CA3, CA4, k), dtd^p(CA3, CA4, k) \rangle \\
& td^p(CA3, CA4, k) = 0.866; \\
& dtd^p(CA3, CA4, k) = 0.037; \\
& ud^p(CA3, CA4, k) = 0.097;
\end{aligned} \tag{60}$$

the derived trust relationship from Alice to CA4 is:

$$\begin{aligned}
& tr^p(Alice, CA4, k) \\
& = \langle td^p(Alice, CA4, k), dtd^p(Alice, CA4, k) \rangle \\
& td^p(Alice, CA4, k) = 0.849; \\
& dtd^p(Alice, CA4, k) = 0.045; \\
& ud^p(Alice, CA4, k) = 0.106,
\end{aligned} \tag{61}$$

and Alice may accept this level of trust to Bob's public key.

This example shows when quantified risk is introduced in certificate chains, the most trustworthy certification path with respect to "issuing and maintaining certificates", need not be the shortest path.

Practical application of the calculus here might be to provide a framework for accepting or rejection a validated. The risk is that one or more CA's in the chain may have erroneously bound an identity and public key, allowing for subversion of the binding of Bob's identity and the public key in his certificate. If some chain chosen (e.g. the shortest) yields an unacceptably low level of trust, then another chain may be sought and validated, in an effort to find a chain with a high enough trust value. It is important to note here that the different paths through CAs are disjoint and hence statistically independent. In particular, knowledge that all the signatures on one path "checked out" in no way influences the probability that the signatures on another path will, because only one CA signs Bob's certificate. The situation changes significantly though when multiple CAs sign a certificate.

7.2.2 Using multiple-paths certification

Inspired by network reliability, Reiter and Stubblebine's research [35] [36] proposed "resilient authentication" by using redundant multiple independent paths to increase assurance or reliability. The authors suggest two types of independence: (1) node-disjoint paths with bounded length; and (2) k -connective paths with bounded length, in which to break all path, k nodes have to be removed. By their approach, one misbehaving node (CA) will compromise at most one path. So, in the context of public key certification validation, multiple certification paths will make certificate validation more reliable. The drawback is that there is no quantified trust evaluation on certificates or certification authorities, and there is an implicit assumption that the risk

level of each certificate is the same. By combining this approach with our calculus of trust, a better risk evaluation can be made. We discuss this method as follows.

First, use Reiter and Stubblebine's BDP algorithm [35] to get a set of node-disjoint paths of bounded length. Assume that the totally number of such node-disjoint paths is k .

Second, for each path, use sequence aggregation to calculate the aggregated trust in each path.

The aggregated result is a triple $\langle td, dtd, ud \rangle$. Consider the semantics of these degrees. td represents the conditional probability of the trust anchor believing that the certificate made by the target, given fact that the certificate is made by the target; dtd is the conditional probability of disbelief in the certificate; and ud is the degree of uncertainty in current information status. When more relevant information becomes available and the uncertainty is resolved, the ud will partly go to belief, and partly go to disbelief. In the extreme cases, ud completely goes to belief or disbelief. Consequently this triple can be regarded as a probability interval $[td, td + ud]$, (and now, $td + dtd = 1$.)

Assume that the i^{th} path has probability of p_i being valid, and the aggregated trust in the i^{th} path is $\langle td_i, dtd_i, ud_i \rangle$. Then, we have

$$td_i \leq p_i \leq td_i + ud_i. \tag{62}$$

Since those paths are node-disjoint, they are statistically independent. So, the probability of this public key, certificated by k multiple parallel certification paths, being valid is,

$$p = 1 - \prod_{i=1}^n (1 - p_i) \tag{63}$$

Because each p_i has a range, that probability also has a lower bound and upper bound, i.e.

$$1 - \prod_{i=1}^n (1 - td_i) \leq p \leq 1 - \prod_{i=1}^n (1 - (td_i + ud_i)) \tag{64}$$

Returning to the example, BDP algorithm outputs two paths: CA3 -> CA5 -> CA4, and CA3 -> CA1 -> CA2 -> CA4. For each of them, the trust calculated through sequence aggregation is as follows.

For path 1: CA3 -> CA5 -> CA4,

$$\begin{aligned}
& tr^p(CA3, CA4, k) \\
& = \langle td^p(CA3, CA4, k), dtd^p(CA3, CA4, k) \rangle \\
& td^p(CA3, CA4, k) = 0.488; \\
& dtd^p(CA3, CA4, k) = 0.188; \\
& ud^p(CA3, CA4, k) = 0.324;
\end{aligned} \tag{65}$$

the interval of the probability that CA4's certificate for Bob's public key being valid is [0.488, 0.812], obviously, which is very uncertain; for path 2: CA3 -> CA1 -> CA2 -> CA4,

$$\begin{aligned}
& tr^p(CA3, CA4, k) \\
& = \langle td^p(CA3, CA4, k), dtd^p(CA3, CA4, k) \rangle \\
& td^p(CA3, CA4, k) = 0.866; \\
& dtd^p(CA3, CA4, k) = 0.037; \\
& ud^p(CA3, CA4, k) = 0.097;
\end{aligned} \tag{66}$$

the interval of the probability that CA4's certificate for Bob's public key being valid is [0.866, 0.963].

So, by using these two paths, the probability that CA4's certificate for Bob's public key being valid has a lower bound of

$$1 - (1 - 0.488) \cdot (1 - 0.866) = 0.931, \quad (67)$$

and has an upper bound of

$$1 - (1 - 0.812) \cdot (1 - 0.963) = 0.993. \quad (68)$$

So, by using multiple paths, the interval of the probability of the certificate being valid is [0.931, 0.993].

This example shows that using multiple paths for certification is much more certain and more reliable than using single certification path for validation.

The above multiple paths certification has a drawback that to derive the trust in the target, for each CA in a path, only the trust from the proceed CA to this CA is taken into account, and other CAs' trust relationships to this CA are not considered. For example, in path CA3 -> CA5 -> CA4, to evaluate trust in CA5, only CA3's trust in CA5 is considered; CA1's opinion on CA5 is completely neglected. However, CA3's opinion may be based on a small number of encounters between them; CA1's opinion may be based on a much greater number of encounters.

From the perspective of trust in social networks, to avoid the above possible bias in trust evaluation, using the trust relationships in a network of certificates will make the trust anchor get more opinions about a CA from this CA's neighbor CAs. In this way, the trust anchor can be more objectively evaluate this CA.

Return to the example story. Assume that the trust from CA1 to CA5 is

$$\begin{aligned} tr^b(CA1, CA5, k) &= \langle td^b(CA1, CA5, k), dtd^b(CA1, CA5, k) \rangle \\ td^b(CA1, CA5, k) &= 0.91; \\ dtd^b(CA1, CA5, k) &= 0.02; \\ ud^b(CA1, CA5, k) &= 0.07, \end{aligned} \quad (69)$$

and the opinion is based on 5,000 encounters; CA3's direct trust in CA5 is based on just 100 encounters. Then by using parallel aggregation, the aggregated trust from CA3 to CA5 will be

$$\begin{aligned} tr^b(CA3, CA5, k) &= \langle td^b(CA3, CA5, k), dtd^b(CA3, CA5, k) \rangle \\ td^b(CA3, CA5, k) &= 0.887; \\ dtd^b(CA3, CA5, k) &= 0.033; \\ ud^b(CA3, CA5, k) &= 0.08, \end{aligned} \quad (70)$$

which is significantly different from CA3's direct trust in CA5 (formula 55).

In the following, we briefly discuss how to use the trust relationships in a network of certificates, to evaluate the trust in a CA; and leave the detailed algorithm design and analysis for future research.

For each concerned CA, the trust from the trust anchor can be evaluated by using the algorithm given in section 6.3; then use the derived trust as heuristic information in Reiter and stubblebine's multiple independent certification paths discovery.

Ideally, for trust evaluation in a social network, the more information is used, the more accurate the evaluation is.

It would be perfect to use complete information (all relevant trust relationships) in the network. However in the real world, a social network usually is huge. Consider the cost for computing, it is unreal to use all information. By Simon's theory of *bounded rationality*, a decision making process in the real world is limited by bounded rationality i.e. the "rational choice that takes into account the cognitive limitations of the decision maker - limitations of both knowledge and computational capacity". Thus, it is acceptable to use just partial but most relevant information (trust relationships) to make trust evaluation in a huge social network.

8. CONCLUDING REMARKS

In order to explicitly represent trust and to quantify the risk associated with trust in public key infrastructure (PKI) and identity management (IdM), we introduced a formal semantics based calculus of trust, and demonstrated how to apply the trust calculus, to formally represent trust relationships and to quantitatively evaluate the risk associated with trust in public key certificate chains. This research shows that after introducing formal representation and quantification of trust in certificate chains, for using one-path certification, the shortest certification path need not be the most trustworthy certification path, and that a chain with an acceptably high level of trust should be constructed for validation; for using multi-path certification, multiple independent certification paths provides much more reliable and certain public key certification validation.

To continue the work presented in this paper, the future work can go further in several directions. First, knowledge that a certificate has been validated by some path (or some set of paths) clearly impacts the probability that the certificate will be validated by another. This is a feature of the analysis yet to be developed. Other future work is to develop effective and efficient trust aggregation algorithms in huge size social networks; use trust calculus as heuristic information in public key certification path building, and other applications of the calculus of trust, for example, modeling trust in privacy protection in healthcare.

9. ACKNOWLEDGMENTS

The US Department of Homeland Security, through grant award number 2006-CS-001-000001 under the auspices of the Institute for Information Infrastructure Protection (I3P) research program, partly supported this work. The views and conclusions contained in this document are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of the US Department of Homeland Security, the I3P, or Dartmouth College, which manages the I3P program.

10. REFERENCES

- [1] K. Blomqvist. The many faces of trust. *Scandinavian Journal of Management*, 13(3):271-286, 1997.
- [2] D. Bodeau. *Sharing Protected Identity and Credential Information (SPICI) Framework for Assessable Identity and Privacy Protection*. The MITRE Corporation, 2008.
- [3] W. Burr. Public key infrastructure (pki) technical specifications: Part a - technical concept of operations. 1998.

- [4] M. Burrows, M. Abadi, and R. Needham. A logic of authentication. *ACM Transactions on Computer Systems*, 8(1):18–36, 1990.
- [5] C. Dellarocas and P. Resnick. Online reputation mechanisms a roadmap for future research. In *First Interdisciplinary Symposium on Online Reputation Mechanisms*, 2003.
- [6] R. Demolombe. To trust information sources: a proposal for a modal logical framework. In C. Castelfranchi and Y.-H. Tan, editors, *Trust and Deception in virtual societies*, pages 111–124. Kluwer Academic Publishers, 2001.
- [7] L. Ding, P. Kolari, S. G. Finin, A. Joshi, Y. Peng, and Y. Yesha. Modeling and evaluating trust network inference. In *The Seventh International Workshop on Trust in Agent Societies, at AAMAS 2004*, 2005.
- [8] C. Ellison and B. Schneier. Ten risks of pki: what you're not being told about public key infrastructure. *Computer Security Journal*, XVI(1), 2000.
- [9] Y. Elley, A. Anderson, S. Hanna, S. Mullan, R. Perlman, and S. Proctor. Building certification paths: Forward vs. reverse. In *The 10th Annual Network and Distributed System Security Symposium*, 2001.
- [10] R. Forno and W. Feinbloom. Pki: A question of trust and value. *Communication of the ACM*, 44(6), June 2001.
- [11] J. A. Golbeck. *Computing and Applying Trust in Web-Based Social Networks*. Ph.D. Thesis, University of Maryland, College Park, 2005.
- [12] R. Guha and R. Kumar. Propagation of trust and distrust. In *WWW2004*, 2004.
- [13] A. Hajek. Probability, logic, and probability logic. In L. Goble, editor, *Philosophical Logic*. Blackwell Publishing, 2001.
- [14] S. Hanna and P. Hesse. Approaches to certificate path discovery (slides). In *PKI'2004*, 2004.
- [15] J. Huang. *Knowledge Provenance: An Approach to Modeling and Maintaining The Evolution and Validity of Knowledge*. PhD Thesis, University of Toronto, <http://hdl.handle.net/1807/11112>, December 2007.
- [16] J. Huang and M. S. Fox. An ontology of trust – formal semantics and transitivity. In *Proceedings of The Eighth International Conference on Electronic Commerce*, pages 259–270. ACM, 2006.
- [17] J. Huang and D. Nicol. A formal semantics based calculus of trust. In *ITI Research Report, (to be submitted for journal publication)*, University of Illinois at Urbana-Champaign, 2008.
- [18] A. Josang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness, and Knowledge-Based Systems*, 9(3):279–311, 2001.
- [19] A. Josang, E. Gray, and M. Kinatader. Simplification and analysis of transitive trust networks. *Web Intelligence and Agent Systems Journal*, 4(2):139–161, 2006.
- [20] S. D. Kamvar, M. T. Schlosser, and H. Garcia-Molina. The eigentrust algorithm for reputation management in p2p networks. In *WWW '03: Proceedings of the 12th international conference on World Wide Web*, pages 640–651, New York, NY, USA, 2003. ACM.
- [21] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.
- [22] R. Levien. *Attack-resistant trust metrics*. PhD Thesis, University of California, Berkeley, USA, 2002.
- [23] J. Linn. Trust models and management in public key infrastructures. *RSA Laboratories*, 2000.
- [24] N. Luhmann. *Trust and Power*. John Wiley & Sons Ltd, 1979.
- [25] S. P. Marsh. *Formalising Trust as a Computational Concept*. Ph.D. Thesis, University of Stirling, 1994.
- [26] S. P. Marsh and M. R. Debben. Trust, untrust, distrust and mistrust - an exploration of the dark(er) side. In *Proceedings of iTrust2005, LNCS 3477*, pages 17–33, 2005.
- [27] U. M. Maurer. Modelling a public-key infrastructure. In *ESORICS '96: Proceedings of the 4th European Symposium on Research in Computer Security*, pages 325–350, London, UK, 1996. Springer-Verlag.
- [28] R. Mayer, J. Davis, and F. Schoorman. An integrative model of organizational trust. *Academic of Management Review*, 20(3):709–734, 1995.
- [29] Microsoft. Pki trust models (slides). 2004.
- [30] T. Moses. Pki trust models.
- [31] L. Mui and A. Halberstadt. A computational model of trust and reputation. In *Proc. 35th Hawaii Int. Conf. on System Sciences*, 2002.
- [32] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. *Technical report, Stanford Digital Library Technologies Project*, 1998.
- [33] R. Perlman. An overview of pki trust models. *IEEE Network*, 13:38–43, 1999.
- [34] W. T. Polk and N. E. Hastings. Bridge certification authorities: Connecting b2b public key infrastructures. 2000.
- [35] M. K. Reiter and S. G. Stubblebine. Resilient authentication using path independence. *IEEE Trans. Comput.*, 47(12):1351–1362, 1998.
- [36] M. K. Reiter and S. G. Stubblebine. Authentication metric analysis and design. *ACM Trans. Inf. Syst. Secur.*, 2(2):138–158, 1999.
- [37] C. Satizabal, R. Paez, and J. Forne. Pki trust relationships: from a hybrid architecture to a hierarchical model. 2006.
- [38] B. Yu and M. Singh. A social mechanism of reputation management in electronic communities. In *Proceedings of Fourth International Workshop on Cooperative Information Agents*, pages 154–165, 2000.
- [39] J. Zhang and R. Cohen. Trusting advice from other buyers in e-marketplaces: the problem of unfair ratings. In *Proceedings of The Eighth International Conference on Electronic Commerce*, pages 225–234. ACM, 2006.
- [40] M. Zhao and S. W. Smith. Modeling and evaluation of certification path discovery in the emerging global pki. In *EuroPKI2006*, 2006.