Towards a Dynamic and Composite Model of Trust

Adam J. Lee
Department of Computer Science
University of Pittsburgh
Pittsburgh, PA 15260
adamlee@cs.pitt.edu

Ting Yu
Department of Computer Science
North Carolina State University
Raleigh, NC 27695
tyu@ncsu.edu

ABSTRACT

During their everyday decision making, humans consider the interplay between two types of trust: vertical trust and horizontal trust. *Vertical trust* captures the trust relationships that exist between individuals and institutions, while *horizontal trust* represents the trust that can be inferred from the observations and opinions of others. Although researchers are actively exploring both vertical and horizontal trust within the context of distributed computing (e.g., credential-based trust and reputation-based trust, respectively), the specification and enforcement of composite trust management policies involving the flexible composition of *both* types of trust metrics is currently an unexplored area.

In this paper, we take the first steps towards developing a comprehensive approach to composite trust management for distributed systems. In particular, we conduct a use case analysis to uncover the functional requirements that must be met by composite trust management policy languages. We then present the design and semantics of CTM: a flexible policy language that allows arbitrary composition of horizontal and vertical trust metrics. After showing that CTM embodies each of the requirements discovered during our use case analysis, we demonstrate that CTM can be used to specify a wide range of interesting composite trust management policies, and comment on several systems challenges that arise during the composite trust management process.

Categories and Subject Descriptors: C.2.4 [Distributed Systems]: Distributed applications; D.4.6 [Operating Systems]: Security and Protection—*access controls, authentication*; K.6.5 [Management of Computing and Information Systems]: Security and Protection

General Terms: Security

Keywords: Credentials, policy, reputation, trust

1. INTRODUCTION

The Internet and the World Wide Web enable the formation of large-scale decentralized systems where entities from different security domains can interact in an ad-hoc manner. Such decentralized systems can dramatically improve the flexibility, variety, and convenience of e-services. However, since transactions might occur between entities with no a priori knowledge of one another, the ability to assess the trustworthiness of an entity is of central importance.

Social scientists have identified two types of trust that affect human interactions in a society: vertical trust and horizontal trust [17]. *Vertical trust* captures the trust relationships that exist between individuals and institutions, while *horizontal trust* represents the trust that can be inferred from the observations and opinions of others. These notions of trust are complementary and are often used in concert during everyday decision making. For example, when an individual wishes to make a dinner reservation, she may consider awards and certifications given to potential restaurants by local or national organizations (vertical trust), as well as the experiences of her friends (horizontal trust). Similarly, when deciding whether to hire a new employee, companies evaluate the academic and professional pedigree of each applicant (vertical trust), as well as the opinions of recommendation letter writers, interviewers, and members of the hiring committee (horizontal trust). As the digital world can be viewed as a logical extension of human society, researchers have naturally developed both horizontal and vertical trust models for use in distributed systems.

Historically, the vast majority of trust assessment mechanisms used in distributed systems have relied on vertical trust. Role-based access control (RBAC) [30], attribute-based access control (ABAC) [34], federated identity [27,32], trust negotiation [4, 5, 36], and distributed proof [3, 19] systems all assess the trustworthiness of a user based upon verifiable and unforgeable statements regarding the identity, roles, or attributes ascribed to that user by one or more trusted certifiers. That is, these trusted certifiers serve as "roots" of vertical chains of trust associated with a particular principal. More recently, the popularity of peer-to-peer systems and social networks has spawned great interest in the design and deployment of horizontal trust mechanisms for many different application domains. Such mechanisms include reputation/recommendation systems (e.g., [14, 15]), collaborative filtering applications (e.g., [1, 29]), and other history-based audit mechanisms (e.g., [2, 28]). These systems allow the observations of the many to be aggregated and processed according to a given principal's specifications, thereby establishing a transitive notion of horizontal trust.

Much work has been done on both types of trust models, and each has both advantages and limitations. Credential-

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.
SACMAT'09, June 3–5, 2009, Stresa, Italy.
Copyright 2009 ACM 978-1-60558-537-6/09/06 ...$5.00.

based trust is suitable for expressing the pre-conditions for a trustworthy party. For example, Alice may specify that a service provider has to be member of BBB and be in business for more than three years before she will consider using its service. Furthermore, vertical trust establishment systems typically have a well-defined formal semantics and somewhat rigid structure to ensure that they remain amenable to formal analysis. However, credential-based trust is binary: i.e., a principal is either trusted or untrusted. On the other hand, horizontal trust establishment systems are more flexible, as they can reflect trust inferred from a principal's past behavior. Unfortunately, due to their open nature, these types of systems can often be subjected to manipulation by attackers (e.g., see [18]).

As computer systems begin to more closely mimic the physical world, significant benefit could be realized by combining horizontal (e.g., reputation-based) and vertical (i.e., credential-based) trust assessment mechanisms for distributed systems. Unfortunately, this integration of horizontal and vertical trust within the digital world has not yet occurred. Very few research efforts have explored the combination of these two types of trust, and existing proposals support only simple conjunction and disjunction of horizontal and vertical trust metrics [6, 10]. Although this approach is a step in the right direction and can express simple policies such as *I will install an application only if its author is a member of the BBB and has a reputation of at least 0.85*, it is insufficient for expressing many realistic policies involving sequential composition. For instance, simple conjunction and disjunction cannot be used to express the policy *I will install an application only if its author is a member of the BBB and has a reputation of at least 0.85, as reported by members of the ACM.*

In an effort to develop a comprehensive approach to the integration of horizontal and vertical trust metrics in distributed systems, we present a unified trust management policy language called CTM. Unlike existing proposals, CTM allows policy writers to compose horizontal and vertical trust metrics in an arbitrary manner. For example, CTM can handle the types of sequential composition discussed above. Furthermore, CTM policies can be nested to an arbitrary depth, which allows for the representation of non-trivial decision making processes. This approach to trust management allows users to make trust decisions in the digital world by using the same types of a reasoning that they rely upon in the physical world. In this paper, we make the following contributions:

- We systematically identify the desirable properties for a composite model of trust by examining use cases from the social networking, process automation, and virtual organization domains.
- We develop a unified trust management policy language, CTM, which extends the *RT* family of policy languages [24] to allow arbitrary composition of horizontal and vertical trust metrics.
- We provide a formal semantics for this language based upon the concept of role membership.
- We provide detailed and realistic examples to demonstrate the flexibility and expressiveness of our approach.
- We note that this work represents a first step towards building a composite trust framework that can be deployed and utilized within a variety of application domains. To this end, we also discuss several interesting research issues related to the efficient evaluation of CTM policies in decentralized systems.

Section 2 sets the stage for the rest of this paper by presenting an overview of the characteristics of large-scale open systems. Section 3 explores several use cases in an effort to identify desirable features for any composite trust management policy language. We then present the syntax and semantics of the CTM policy language in Section 4. In Section 5, we show that CTM includes all of the features identified in Section 3 and that it can be used to express a wide range of interesting policies. We also discuss the systems challenges associated with the efficient evaluation of CTM policies. We then highlight relevant related work in Section 6, and present our conclusions and directions for future work in Section 7.

2. SYSTEM MODEL

In the context of our discussion, the *openness* of a computing system mainly concerns its management of trust policies and decisions. Specifically, a system is an open system if the principals involved in the system can make autonomous decisions regarding who can access their resources or provide services to them. This definition of openness does not impose any restrictions on where principals are physically located, how they communicate with each other, what schemes are used to manage trust-related information, or how their trust decisions are enforced. For example, we consider eBay to be an open system, since each user in eBay determines solely by him/herself when it suffices to trust a seller or a buyer based upon their past behavior, even though all such information is centrally managed by a single authority (i.e., eBay). Similarly, decentralized systems with multiple trust authorities (e.g., virtual organizations, computing grids, etc.) are also examples of open systems, as principals in these systems can make autonomous trust decisions on an individual basis.

A principal in an open system forms its trust decisions based on credentials, i.e., statements made by itself or other principals. Different from typical settings in trust management, credentials in our discussion are not limited to digitally signed certificates, but are generalized to refer to any statements that can be authenticated and verified. Such a generalization is important as it opens doors to much broader information sources for trust decisions, including organizational LDAP directories, role databases in RBAC, and audit trails generated by intrusion detection systems. In particular, feedback reports and ratings in online communities and P2P systems are a special form of credentials as well. Our definition of credentials is environment-independent. Credentials can be either centrally managed in a well-defined domain (e.g., a company's LDAP directory) or totally decentralized and dynamic (e.g., information maintained in P2P). Similarly, credentials can either be issued by some well-recognized authorities (e.g., BBB membership or driver's license) or by arbitrary principals who do not have any special roles or responsibilities.

For example, in Figure 1, credentials are taken from multiple sources with different formats. In uSell.com, an online auction site, feedback credentials may be issued by arbitrary users in that domain, though all these credentials are centrally and exclusively managed by uSell.com. Similarly, any-

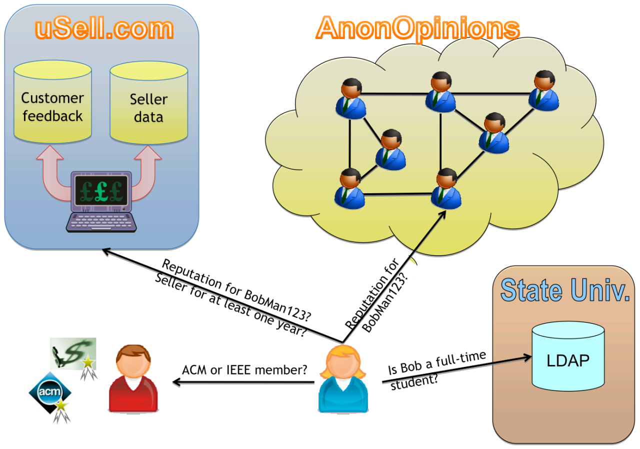
uSell.com
Customer feedback
Seller data
AnonOpinions
State Univ.
LDAP
Reputation for BobMan123?
Seller for at least one year?
Reputation for BobMan123?
ACM or IEEE member?
Is Bob a full-time student?
acm

Figure 1: Example sources of credentials and trust policies

body in AnonOpinions can issue opinion credentials about others. However, different from uSell.com, AnonOpinion is a decentralized P2P system, where opinion credentials are totally distributed among all the principals in the domain. Even for credentials issued by well-known authorities, the way they are stored (and thus retrieved) may be quite different. In Figure 1, ACM or IEEE membership credentials are held by individual members, while student enrollment statuses are centrally maintained by a university's LDAP server.

3. POLICY LANGUAGE REQUIREMENTS

In this section, we first explore applications of composite trust management systems within three different domains. We then use these example applications to derive a set of important features that should be afforded by any composite trust management policy language.

3.1 Motivating Scenarios

In recent years, distributed computing systems have evolved from physically distributed systems operated within a *single* administrative domain into systems that provide service to users from *many* administrative domains. Some of these systems have even begun to approximate the diversity of every day life by tightly integrating with the physical world (e.g., ubiquitous computing) or providing support for basic human interactions in a digital setting (e.g., social networking sites like Facebook). Composite trust management approaches could help system administrators and users of these systems make better access control and resource management decisions by enabling them to apply techniques from day-to-day life to their online interactions.

Existing systems have a variety of information sources that can be used to establish both horizontal and vertical trust. For example, information for establishing vertical trust in an entity can be gathered from digital certificates such as X.509 credentials, LDAP directories operated by trusted organizations, quasi-certified attribute data in social networking profiles, or public databases (e.g., voter registration databases). On the other hand, information gathered from reputation/recommendation systems, informal polls run by various web sites, logs generated by intrusion detection systems, or historical QoS data can be used as a basis for establishing horizontal trust in an entity. We now present three scenarios exploring the uses of composite trust management in a variety of contexts.

Scenario 1: Social Networking

Alice is an avid fan of social networking and regularly uses the site MyFriends.com to keep in touch with friends, meet new people, and play games. Unfortunately, installing a new application to her account requires that Alice reveal all of her profile data to that application. Alice is very security conscious, however, and wants to limit the types of applications that have access to her profile. Specifically, Alice will only install an application if its author can present a TRUSTe-issued privacy policy certificate and has a reputation rating of at least 0.8 (out of a possible 1.0) as calculated by Alice's friends who list "computer security" as one of their interests.

Scenario 2: Process Automation

Acme Incorporated is a large international corporation that runs an online recruiting portal to facilitate the application process. Due to the large number of applications received during any given recruiting cycle, Acme's recruiting portal batches received applications automatically based upon the characteristics of the applicant submitting the recruiting package. Acme wants to define an application category called "priority" for applicants who attended a "preferred" academic institution (as defined in Acme's corporate policy), are members of the ACM or IEEE, and whose average departmental "Black Friday" score is at least 9.0 (out of a possible 10). Further, only the Black Friday scores reported by tenured faculty members should be included when calculating the above average.

Scenario 3: Virtual organizations

The PetaGrid is a national-scale grid computing network that provides computational resources to students, academic faculty, and scientists throughout the United States. The Nation-Wide Center for Computation (NWCC) is a member organization of the PetaGrid. In order to request an allocation on one of their clusters, the NWCC requires that the requestor be either a graduate student or faculty member at an ABET-accredited university or an employee of a DOE or DOD research laboratory. Furthermore, requestors must have a history of using at least 85% of their previous computing allocations, as reported by PetaGrid member organizations that are themselves rated as being at least 90% reliable by at least 10 members of PetaGrid's industrial advisory board. When computing reliability ratings, the influence of each opinion is weighted using an exponential decay function so that current history outweighs historical evidence.

Each of the above application scenarios makes use of digital credentials and other, more subjective, horizontal trust metrics during the decision making process. However, despite their similarities, these scenarios also differ from one another in very significant ways. These similarities and differences imply that constructing a comprehensive approach to composite trust management requires the design of a flexible policy language capable of expressing a wide range of policies, as well as technical consideration of the non-trivial systems issues associated with gathering the evidence needed for policy enforcement (which we discuss in Section 5.3). In the remainder of this paper, our primary focus lies in the development of a policy language suitable for use in composite trust management systems.

3.2 Desiderata

An examination of the above scenarios has led us to identify five requirements for composite trust management policy languages, which go beyond the more general requirements for trust negotiation and trust management policy languages described in [31]. We now briefly describe each of these requirements.

Platform Neutrality. A composite trust management system should support the use of centralized attribute repositories, as well as decentralized attribute credentials. It should also be able to make use of both centralized and decentralized sources of horizontal trust data. For example, in the social networking scenario, the "interests" maintained in a user's profile are centrally managed by the social networking site, while an application author's TRUSTe-issued privacy policy would be managed by the author as a digital certificate. Note also that the "Black Friday" rating calculated in the process automation scenario would likely be computed at a central server managed by the student's home department. By contrast, the percent utilization calculation performed in the virtual organization scenario would be best carried out in a decentralized manner by querying utilization histories stored across multiple PetaGrid nodes.

Algorithmic Flexibility. A composite trust management system should support the use of an arbitrary set of aggregation algorithms for horizontal trust data. For instance, the reputation rating calculated in the social networking scenario is nothing more than a simple average of individual user ratings, while the reliability calculation in the virtual organization scenario uses a weighted average in which the influence of old scores decays with age. Similarly, a user may wish to download feedback reports from, e.g., the eBay feedback database, but combine these reports using his or her own algorithm. The ability to support an extensible array of aggregation algorithms will help ensure the wide-applicability of the language.

Unified Representation. Composite trust management policy languages should represent the result of a horizontal trust assessment calculation as a first-class digital credential. In each of the preceding scenarios, the English-language policy that is specified makes no distinction between the digital credentials that must be presented and the horizontal trust assessment calculations that must be carried out to satisfy the policy. That is, the result of each such calculation is treated as a piece of evidence that is calculated in a trusted manner and attests to some attribute of a particular principal. Languages satisfying this requirement would allow, e.g., reputation data to be included into a policy without introducing new policy composition operators, and would allow fields within the calculated result to be constrained in the same manner as fields within a digital credential.

Flexible Composition. Most policy languages designed for use in decentralized systems—such as XACML, Cassandra [4], and *RT* [24]—allow policies to be defined in terms of other sub-policies. This is quite advantageous, as sub-policies can be reused throughout an organization and complex policies can be specified in a very natural manner by composing simpler sub-policies. The example scenarios presented above extend this notion into the domain of composite trust management systems by using sub-policies to filter both the providers of digital credentials (e.g., "preferred" academic institutions), as well as the input providers for reputation or rating information (e.g., Black Friday scores are collected only from tenured faculty members). Furthermore, the virtual organization scenario requires multiple levels of sub-policy to constrain the calculation of a user's percent utilization score.

Declarative Semantics. The policy language used to specify a composite trust management policy should have a declarative semantics. This ensures that every policy has a *precise* meaning that is completely decoupled from the policy's enforcement algorithm. In addition to enabling thorough formal analysis of security policies, this requirement also makes it possible for multiple policy enforcement and evaluation implementations to interoperate with one another in a straightforward manner.

We now describe the syntax and semantics of the CTM composite trust management policy language. In Section 5, we will revisit the scenarios discussed in this section to show that CTM embodies each of the above requirements.

4. A COMPOSITE TRUST MODEL

Vertical and horizontal trust are quite different in terms of their definition and evaluation. Vertical trust is often defined through logical inference rules, while horizontal trust is achieved through different types of aggregation. To enable the flexible composition of these concepts, the key is to come up with a unified semantics for interpreting, and thus bridging, both types of trust. This will provide a "closure property" that allows the combination of two or more trust policies to create another policy that can be interpreted using the same semantics concept. Through this process, policy writers are able to build complicated trust policies gradually from simpler ones, and consolidate policies defined by principals in different domains.

Many vertical trust management systems, such as SPKI/SDSI and *RT*, use the concept of role membership to define a set-based semantics for policies. Intuitively, a role defines a set of principals possessing the same properties, while a policy defines a set of role memberships that must be be possessed by an authorized principal. As a result, policies and roles can be composed using standard set operations (union, intersection, etc.) and deciding whether a principal satisfies some policy becomes a set membership test. In this section, we show that this set-based policy semantics can be extended to include roles defined in terms of horizontal trust metrics, such as reputation scores. This implies that vertical trust management policy languages can be extended to allow arbitrary composition of roles defined in terms of both horizontal and vertical trust metrics *without* introducing new policy composition operators. To this end, we present CTM, which extends the *RT* family of trust management languages to support composite trust management. We begin with an overview of the *RT* languages.

4.1 RT_0 and RT_1 Policy Syntax

RT_0 is the most basic language in *RT* family of trust management languages. As in all of the *RT* languages, principals are identified by means of identity certificates. RT_0 roles are defined simply as strings identifying the name of the role and cannot be parameterized (i.e., roles cannot contain internal attributes). Policy statements in RT_0 are expressed as one or more of these role definitions and are encoded as role definition credentials signed by the author of the role definition. There are four basic types of role definitions in RT_0:

Simple Member. A role definition of the form $K_A.R \leftarrow K_D$ encodes the fact that principal K_A considers principal K_D to be a member of the role $K_A.R$. That is, $K_D \in K_A.R$.

Simple Containment. A role definition of the form $K_A.R \leftarrow K_B.R_1$ encodes the fact that principal K_A defines the role $K_A.R$ to contain all members of the role $K_B.R_1$, which is defined by principal K_B. That is, $K_B.R_1 \subseteq K_A.R$.

Linking Containment. A role definition of the form $K_A.R \leftarrow K_A.R_1.R_2$ is called a linked role. This defines the members of $K_A.R$ to contain all members of $K_B.R_2$ for each K_B that is a member of $K_A.R_1$. That is, $\{p \mid p \in K_B.R_2 \wedge K_B \in K_A.R_1\} \subseteq K_A.R$.

Intersection Containment. The role definition $K_A.R \leftarrow K_{B_1}.R_1 \cap \cdots \cap K_{B_n}.R_n$ defines $K_A.R$ to contain the principals who are members of each role $K_{B_i}.R_i$ where $1 \leq i \leq n$. That is, $K_{B_1}.R_1 \cap \cdots \cap K_{B_n}.R_n \subseteq K_A.R$.

These four basic types of role definitions can be used to define a wide range of access control policies. For example, the following RT_0 role definitions express an access control policy requiring that entities accessing a given resource be employees of a PetaGrid member organization:

$$\begin{gathered} \mathit{Provider.service} \leftarrow \mathit{Provider.partner.employee} \\ \mathit{Provider.partner} \leftarrow \mathit{PetaGrid.memberOrganization} \end{gathered}$$

If a principal, Alice, could provide credentials proving the statements $\mathit{PetaGrid.memberOrganization} \leftarrow \mathit{AliceLabs}$ and $\mathit{AliceLabs.employee} \leftarrow \mathit{Alice}$, she could satisfy the policy formed by the above two role definitions and gain access to the protected service.

RT_1 extends RT_0 by adding the ability to parameterize role definitions. For example, consider the following two RT_1 role definitions:

$$\begin{gathered} \mathit{AliceLabs.employee}(\mathit{title} = \text{“President”}) \leftarrow \mathit{Alice} \\ \mathit{Acme.sale} \leftarrow \mathit{Acme.widget}(\mathit{price} > 10) \end{gathered}$$

The first role definition declares that Alice is not only an employee of AliceLabs, but also that she is the President of AliceLabs. The second role definition says that Acme's "sale" role contains all members of the widget role whose "price" attribute is greater than \$10. We now explore how this notion of roles and its associated set-based semantics can be extended to support horizontal trust assessment functions based upon aggregated data.

4.2 Assumptions Regarding Horizontal Trust

We assume that the set $\mathcal{P}$ of principals in an open system interact through a series of transactions such as file downloads, supercomputer utilization within a virtual organization, or information retrieval from a website. After a transaction has completed, the principals involved in the transaction may issue feedback reports to rate each other's behavior. A feedback report may include multiple properties of a transaction such as its time, its type (e.g., downloading a file, reading an article, or winning an auction), and its volume (e.g., the size of a file or the amount of money paid). Similarly, each feedback report may rate multiple aspects of a single transaction (e.g., product quality, customer service, and timeliness of delivery). Without loss of generality, in our discussion we assume that a feedback contains the following properties: (1) the issuer, i.e., the entity who generates the feedback; (2) the subject, i.e., about whose behavior this feedback applies to; (3) the signer, i.e., who certifies the feedback; (4) a single rating; and (5) other transaction-specific properties (e.g., the transaction type, the time of the transaction, the data volume for file downloading transactions, etc). We denote by $\mathcal{F}$ the set of all feedback reports. Adding support for additional rating dimensions is a trivial extension to the model presented in this paper, and is omitted for simplicity.

Unlike standard digital credentials (e.g., X.509 certificates), we differentiate the issuer and the signer of a feedback score to better accommodate various forms of digital statements. For example, a feedback report at an online auction site is typically issued by a buyer about a seller for a particular transaction. However, this feedback report is certified (either digitally signed or provided securely) by the site instead of by the issuer. Note that this is different from delegations. The auction site does not speak for the buyer, but simply certifies the fact that the buyer issued this feedback report about the seller.

We also note that the boundary between feedback reports (for horizontal trust) and other credentials (for vertical trust) may not be all that clear. In particular, a feedback report may be used in both vertical and horizontal trust decisions. For example, considering the restaurant ratings from local magazines. One may trust a restaurant as long as it is rated with a score more than 90 by any local magazine or by a particular magazine. In this case, a magazine's rating is considered to be issued by some authority. Meanwhile, another user may trust a restaurant only if the average rating of all the magazines is over 90, which is closer to a horizontal trust decision.

4.3 Supporting Horizontal Trust

Intuitively, a horizontal trust assessment function (e.g., a reputation function) takes a set of feedback reports and computes a trust metric for some principal (called the *target* of an evaluation). Many horizontal trust metrics are subjective in nature. That is, given the same set of feedback reports, a principal may be assigned different trust values by different principals, even if the same trust function is used. Thus, when applying a horizontal trust assessment function, we also need to identify from whose point of view the target's trust value is computed. We call this entity the *source* of a horizontal trust evaluation. Formally, such a calculation can be modeled as a function $f : 2^{\mathcal{F}} \times \mathcal{P} \times \mathcal{P} \rightarrow \mathcal{R}$, where $2^{\mathcal{F}}$ is the power set of $\mathcal{F}$, and $\mathcal{R}$ is reputation domain (e.g., the interval $[0, 1]$). Given a set of feedback reports, and source and target principals, f returns a reputation score.

The above definition makes the simplifying assumption that a horizontal trust evaluation function takes no parameters other than the source and target principals, and the set of feedback reports to consider. In practice, such functions often do have additional parameters, such as constants used to control the weight assigned to a particular feedback based upon its age, or explicit limits on the number of feedback reports to be collected. For clarity of presentation, we omit such additional parameters from our discussion, and note that including them is a trivial extension to CTM.

Recall that roles are typically used to group principals with similar attribute characteristics (e.g., the same employer, etc.). We observe that the assessment reported for a given principal after invoking a horizontal trust function is effectively an attribute of that user. As such, the concept of a parameterized role can be used to group principals with similar horizontal trust characteristics. To enable the arbitrary composition of horizontal and vertical trust metrics, we define CTM as a role-based policy language that allows roles to be defined using the four role types defined by the RT_1 language (see Section 4.1), as well as by using the following concept of an *aggregate containment role*:

Aggregate Containment. Let $\diamond$ represent a comparison operator (e.g., $<, \leq, =, \geq, >$, or $\neq$). The role definition $K_A.R \leftarrow K_B.F(issuer = K_i.R_i, target = K_t.R_t, signer = K_s.R_s, rating \diamond c_r, a_1 \diamond c_1, \ldots, a_n \diamond c_n, output \diamond c_o)$ defines the role $K_A.R$ to contain all principals whose horizontal trust level satisfies the constraint $output \diamond c_o$ after principal K_B invokes the horizontal trust assessment function F when considering feedback reports satisfying the constraint $rating \diamond c_r$ issued by principals in the role $K_i.R_i$ referring to targets in the role $K_t.R_t$ that are signed by signers in the role $K_s.R_s$. Each a_k represents an additional attribute of the reputation report that can also be constrained.

We can extend the set-based semantics of RT to the notion of aggregate containment. Specifically, we note that based upon the above definition, the relationship $\{p \in \mathcal{P} \mid F(R, K_B, p) \diamond c_o \wedge r \in R \rightarrow (r.issuer \in K_i.R_i \wedge r.target \in K_t.R_t \wedge r.signer \in K_s.R_s \wedge r.rating \diamond c_r \wedge r.a_1 \diamond c_1 \wedge \cdots \wedge r.a_n \diamond c_n)\} \subseteq K_A.R$ holds. This unified semantics enables the composition of role definitions based upon vertical and horizontal trust metrics can occur without the need to introduce new policy composition operators.

4.4 Arbitrary Composition

To demonstrate that CTM allows policies to be composed in a truly arbitrary fashion, we must show that CTM supports three basic types of policy (strictly vertical policies, strictly horizontal policies, and policies involving simple conjunction of horizontal and vertical trust metrics), as well as policies involving sequential composition of horizontal, vertical, and hybrid roles. We first consider the three basic policy types. Clearly, the RT_1 language is used to specify policies involving strictly vertical trust metrics. Since CTM provides a superset of the functionality offered by RT_1, it is capable of specifying this type of policy as well. Similarly, any policy specified using only the aggregate containment, simple containment, and intersection containment rules is a strictly horizontal policy, as membership can only be defined in terms aggregations over feedback reports. By also allowing the use of the simple member rule, policies involving simple conjunctions of horizontal and vertical trust metrics can be supported by CTM.

Sequential composition of roles occurs when one role is applied as a filter to limit the input set considered by another role. The linking containment $K_A.R \leftarrow K_A.R_1.R_2$ expresses the sequential composition in which the role $K_A.R_1$ is used to limit the allowable certifiers of membership in the R_2 role. Within CTM, $K_A.R_1$ can be either a horizontal, vertical, or hybrid (i.e., involving both horizontal and vertical trust) role, any type of role can be used to limit the certifiers of the vertical trust relationship encoded by the R_2 role. Similarly, the K_i, K_s, and K_t roles used to constrain the issuers, signers, and targets considered by an aggregate containment rule can be specified as either horizontal, vertical, or hybrid roles. This implies that any type of role can be used to limit the input set used to determine a horizontal trust relationship. For example, consider the following examples:

$$
\begin{aligned}
K_A.R_1 &\leftarrow K_A.f(output > 0.9) \\
K_A.R_2 &\leftarrow K_A.f(issuer = ACM.member, output > 0.9) \\
K_A.R_3 &\leftarrow ACM.member \cap K_A.R_2 \\
K_A.R_4 &\leftarrow ACM.member \cap \\
&\qquad K_A.f(issuer = K_A.R_3, output > 0.9)
\end{aligned}
$$

If K_A is Alice's public key and f is a horizontal trust assessment function, then the role $K_A.R_1$ is defined to contain all principals whom Alice believes to have a trustworthiness rating of over 0.9. The role $K_A.R_2$ contains all principals whom Alice determines to have a trustworthiness rating of over 0.9, based upon feedback reports issued by ACM members; this requires the sequential composition of horizontal and vertical trust. The role $K_A.R_3$ contains the members of $K_A.R_2$ who are themselves ACM members. Finally, $K_A.R_4$ is defined using multiple levels of policy composition to contain all ACM members whom Alice believes to have a trustworthiness rating of over 0.9 based upon feedback reports issued by ACM members who themselves have a trustworthiness rating of over 0.9 according to other ACM members.

As an aside, we also note that the internal fields of a aggregate containment role definition can be used to parameterize the head of the rule in which it appears. For example, consider the following role definition:

$$
\begin{gathered}
K_B.R_1(trust = output, startYear = 2000) \leftarrow \\
K_B.F(issuer = K_B.Friend(since \geq 2000), output \geq 0.8)
\end{gathered}
$$

The definition of the $K_B.R_1$ role preserves both the output of the trust calculation preformed using the function F, as well as the condition on the year parameter of the $K_B.Friend$ role used to filter the set of allowable feedback report issuers. As a result, later role definitions can take these parameters into account to further constrain their memberships. For instance, the role definition $K_B.R_2 \leftarrow K_B.R_1(trustlevel > 0.95)$ contains those members of $K_B.R_1$ whose horizontal trust score is over 0.95. In an implementation of CTM, this may reduce unnecessary calculation by allowing decisions to be made based upon previously computed and cached results.

5. DISCUSSION

In this section, we revisit the scenarios discussed in Section 3.1 to demonstrate that CTM is capable of expressing a range of interesting security policies. These sample policies are then used as a basis for demonstrating that CTM embodies all of the features identified in Section 3.2. We conclude this section with a discussion of implementation challenges

Scenario 1: Social Networking

$$Alice.Trusted \leftarrow TRUSTe.PolicyHolder \cap Alice.Rep(filter = Alice.SecurityPals, output \geq 0.8) \tag{1}$$

$$Alice.SecurityPals \leftarrow MyFriends.Interest(category = \text{“Computer Security”}) \tag{2}$$

Scenario 2: Process Automation

$$Acme.Priority \leftarrow Acme.PrefUniv.Student \cap Acme.ProfOrg.Member \cap Acme.PrefUniv.BlackFri(issuer = Acme.TF, output \geq 9.0) \tag{3}$$

$$Acme.ProfOrg \leftarrow ACM \tag{4}$$

$$Acme.ProfOrg \leftarrow IEEE \tag{5}$$

$$Acme.TF \leftarrow Acme.PrefUniv.Faculty(tenure = \mathsf{True}) \tag{6}$$

$$Acme.PrefUniv \leftarrow StateU \tag{7}$$

Scenario 3: Virtual Organizations

$$NWCC.Auth \leftarrow NWCC.User \cap NWCC.Utilization(issuer = NWCC.Trusted, output \geq 0.85) \tag{8}$$

$$NWCC.User \leftarrow DoD.Employee \tag{9}$$

$$NWCC.User \leftarrow DoE.Employee \tag{10}$$

$$NWCC.User \leftarrow ABET.Accredited.Faculty \tag{11}$$

$$NWCC.User \leftarrow ABET.Accredited.Student(type = \text{“Graduate”}) \tag{12}$$

$$NWCC.Trusted \leftarrow NWCC.Reliable(issuer = PetaGrid.IAB, count \geq 10, output \geq 0.9) \tag{13}$$

Figure 2: CTM policies for each of the scenarios described in Section 3.1.

that will need to be addressed during the design of a comprehensive approach to composite trust management.

5.1 Example Policies

Section 3.1 presented example use cases of composite trust management drawn from the social networking, process automation, and virtual organization domains. These policies are expressed using the CTM policy language in Figure 2, and are easily interpreted by individuals familiar with the *RT* family of policy languages. For clarity of presentation, however, we now explain the process automation policy defined by statements 3–7 of Figure 2. Statement 3 specifies that "priority" applications are students at one of Acme's "preferred" academic institutions, members of a professional organization recognized by Acme, and have a Black Friday score of at least 9.0, as reported by members of Acme's *TF* role. Statements 4 and 5 define the ACM and IEEE, respectively, as professional organizations recognized by Acme. Statement 6 defines Acme's *TF* role to contain any principal capable of proving membership in the *Faculty(tenure = True)* role defined at any one of Acme's preferred universities. Lastly, statement 7 declares that State University is one of Acme's preferred universities.

5.2 Requirements Revisited

Section 3.2 identified five functional requirements that should be met by composite trust management policy languages: platform neutrality, algorithmic flexibility, unified representation, flexible composition, and declarative semantics. As was discussed in Section 4, CTM extends the declarative set-based semantics of *RT*, which has two main effects. First, this clearly satisfies the *declarative semantics* requirement. Secondly, this decouples the enforcement of a policy from its specification. To bridge this gap between specification and enforcement, *RT* relies on the notion of *application domain specification documents* (ADSDs), which specify the internal structure of the role definitions and digital credentials comprising a policy, much in the same way that a database schema describes the contents of an RDBMS table [26]. Within the context of CTM, this notion of ADSDs can be extended also to describe the definition of a particular aggregation containment role, e.g., by describing the (de)centralized nature of the underlying algorithm, and the details of how individual ratings are to be combined. This use of ADSDs therefore allows CTM to satisfy the *platform neutrality* and *algorithmic flexibility* requirements.

The satisfaction of the remaining two requirements follows directly from the way in which aggregation containment is defined in Section 4.3. The *unified representation* requirement is satisfied by CTM because the result of a horizontal trust calculation is treated as a special case of role membership. This allows horizontal trust assessments to appear on the right hand side of any standard *RT* role definition. Furthermore, the internal fields comprising a reputation role can be constrained using any type of CTM role definition. This results in the ability to arbitrarily compose horizontal and vertical trust metrics, as was discussed in Section 4.4. For example, in the virtual organization security policy defined in Figure 2, membership in the *NWCC.Auth* role requires vertical trust (e.g., evidence that the principal is an employee of the DoD), as well as horizontal trust (i.e., evidence that the principal has a history of using at least 85% of his or her allocation). The inputs to this horizontal trust metric are further attenuated by requiring horizontal and vertical trust in the entities supplying input to the algorithm. As a result, CTM satisfies the *flexible composition* requirement.

5.3 Technical Features and Considerations

In this paper, we focus primarily on developing a policy language capable of expressing composite trust management policies involving arbitrary composition of horizontal and vertical trust metrics. However, this is only the first step towards developing a comprehensive approach to composite trust management. In order to fully realize the potential of this approach to trust management, a variety of systems issues must be addressed. We now discuss several such issues to highlight future research challenges.

Correct and Efficient Policy Evaluation. Typically, a trust management system is concerned with answering the *proof of compliance question*: Is there sufficient evidence to determine that some policy P is satisfied? Within the context of a vertical trust management system, answering this question usually involves a search to determine whether a given principal possesses some collection of credentials that satisfy the policy P. In a horizontal trust management system, the solution to this question can again be formulated as a search problem that seeks to determine whether observations recorded in the system (e.g., reputation scores, audit records, etc.) are consistent with the conditions set forth in the policy.

Checking proof of compliance within a composite trust management system will require moving well beyond existing approaches, which make *either* horizontal or vertical trust decisions using *either* centralized or decentralized evidence. Compliance checkers for existing models of trust are often formulated as distributed proof construction procedures that interleave the processes of policy discovery and evaluation in a top-down manner [3, 19]. Although this is an intelligent approach in strictly vertical systems, it breaks down if policies may also invoke horizontal trust assessment algorithms. The question of how to efficiently interleave searches for decentralized credentials with the execution of potentially-expensive horizontal trust assessment algorithms is a non-trivial problem that gives rise to many challenges requiring careful consideration. For example, note that horizontal trust assessment algorithms may incur extremely high overheads if large amounts of distributed evidence (e.g., ratings) need to be discovered, filtered, and/or aggregated. Since a single policy may be satisfiable in several ways, naively invoking these algorithms as the policy is explored can lead to the unnecessary calculation of results that may ultimately be ignored.

Top-k Query Processing. In existing trust management systems, the proof of compliance question is answered in a binary manner: either a policy is satisfied, or it is not. However, because CTM is able to make use of more subjective horizontal trust assessment algorithms, policies can contain a continuous component as well. For example, consider a simple policy P that is satisfied by presenting some collection of digital certificates, as well as a collection decentralized evidence attesting to the fact that the principal in question has historically met certain QoS guarantees. It is not only possible to determine whether two principals satisfy P, but also to compare the degree to which each principal satisfies P (e.g., which principal has provided the best QoS). This ability to rank-order the set of principals satisfying a policy is a novel feature of CTM with many uses, including determining the best set of principals to use when forming a committee, or identifying the most interesting, yet safe, applications to install within the context of a social networking system.

Clearly, the ability to execute top-k queries within a trust management system has interesting applications, although there are many open questions that must be explored. For example, in policies that can be satisfied in more than one way, is taking one path to satisfaction preferable to another? In the case that multiple continuous horizontal trust metrics are used, how is the relative importance of each metric towards the overall trust establishment process weighted? Should the relative importance of a continuous trust metric be dependent upon its height in the generated proof tree? If so, how should this importance be scaled? In addition to these trust assessment questions, there are also many questions related to efficient policy evaluation for collections of principals. For example, how will the potential decentralization of evidence used to calculate horizontal trust metrics impact the evaluation of these metrics for multiple users? In addition to developing evaluation plans that minimize unnecessary computations as was discussed in the proof of compliance case, above, how can we also minimize the communication overhead between principals in the system? In answering these questions, it will be important to identify the situations in which this type of top-k query processing can be efficiently carried out, and those cases in which it cannot.

Coping with Uncertainty and Incomplete Information. In a perfect world, proof of compliance and top-k query processing algorithms would always be guaranteed access to *all* of the evidence required to correctly and completely execute a query. Unfortunately, in any actual system, this is an unrealistic expectation for a number of reasons. For example, users in open systems are autonomous, joining and leaving the network as they please. The resulting churn implies that nodes may be offline at any given time, and thus complete information regarding distributed observations may not be available. In addition to this type of network-imposed limitation, access to data can also be limited from a security perspective. For example, the information that a user requires may exist, but could be protected by one or more *release policies* limiting its disclosure to other principals in the system. Lastly, even if every data provider was simultaneously online and the querier was allowed access to every piece of data, the sheer size of the Internet would make the cost of collecting complete information regarding certain types of distributed observations prohibitive.

The security guarantees afforded by existing trust management systems are largely unaffected by incomplete information. In credential-based vertical trust management systems, the lack of a certain credential results in access being denied to the requester; this fail-safe behavior prevents unsafe decisions from being made in the face of partial information. In horizontal trust management systems (e.g., reputation systems), ratings are not often viewed as security-critical, but rather as suggestions for governing future decisions. As a result, some level of incompleteness is tolerated. However, it is perhaps surprising to note that the arbitrarily composable nature of CTM policies can lead to situations in which incomplete information may negatively affect the policy evaluation process. For example, missing credentials can lead to reduced membership in a role that is later used to filter inputs to a reputation calculation, thereby biasing the results. This bias can affect simple proof of compliance queries, as well as produce unexpected top-k results. In order for composite trust management approaches to be successful, it will be important to investigate metrics for estimating the bias of the observational data collected during a policy evaluation.

6. RELATED WORK

Role-based access control [30] can be viewed as an early type of vertical trust in computing systems, where entities' privileges are decided by the roles they assume. A principal's roles are assigned by a central administrator of a domain. In some sense, an entity's role serves as a virtual credential for it to perform certain actions. Biskup and Karabulut [7] proposed a hybrid model which essentially combines attribute-based access control with traditional capability credentials. This model supports both authorizations and delegations. In this paper, we focus on the composition of aggregate reputation with credentials (including both attribute credentials and capability credentials).

Cross-domain collaboration and resource sharing largely rely on digital credentials for access control. Extensive research has been done in the areas of trust management [8, 24], trust negotiation [36], and distributed proofs [3, 19], which consider both centralized credentials (where credentials are maintained in a well-known repository) and decentralized credentials (where credentials are distributed among multiple entities). In the basic setting of vertical trust management, each entity defines what digital credentials should be presented in order for another entity to access its local resources. The semantics of credentials is greatly enriched to not only include roles but general attributes, which allows very flexible policies suitable for decentralized systems [34]. In the theoretical aspects, researchers investigate appropriate languages for trust policies [19, 24], compliance checking algorithms [9, 26], analysis of policy safety and other properties [25] as well as privacy-preserving negotiation protocols [16]. Researchers have also addressed some of the practical aspects of decentralized vertical trust management, including trust management platform design and implementation [33], and efficient and pragmatic compliance checking algorithms [23].

Reputation-based trust has been studied extensively in the context of agent systems [20]. It has been used as a means to guide the interactions between autonomous agents. The focus is on developing a computing model of reputation that captures how it works in human society. Reputation mechanisms are later deployed in a variety of application domains to facilitate the ad hoc interaction between autonomous entities. Centralized reputation systems where feedback reports and reputation evaluation are managed by a central entity are pervasive nowadays; representative systems include eBay, ePinions, and Amazon.com. Decentralized reputation systems have also been designed for the semantic web, pervasive computing and P2P systems [21, 37]. Early works on reputation systems either assume entities are benign and follow certain behavior pattern consistently or assume simple attack models, which make them subject to easy manipulation. Recently, a large amount of work has been conducted to better understand and countermeasure sophisticated and adaptive attacks [18].

As shown before, the combination of vertical and horizontal trust better captures the trust establishment process used in real life, and is necessary for ensuring that open computing environments (e.g., Web 2.0) provide human users with a realistic means of protecting their personal data. Several research efforts are toward such combination [6, 10]. Unfortunately, these policy languages only support a limited form of combination (e.g., basic conjunction and disjunction). The focus of this paper to develop a trust model that allows highly flexible combination of these two types of trust. Our problem is also related to the problem of policy composition [35]. However, work in the area of policy composition typically only considers how policies of the same type can be combined.

A related area of work involves leveraging user relationships in web-based social networks during online decision making processes. Carminati et al. show that the type, strength, and depth of user trust relationships can be used to control the collection of metadata tags sampled for a given document [12] or to define sets of users that should be allowed to access an online resource [11, 13]. This work is similar in flavor to CTM in that it allows flexible policies to control the collection of recommendation data. However, their systems do not allow for the arbitrary composition of these policies or flexible aggregation and thus cannot represent some of the policies discussed in this paper. Kruk et al. [22] designed an architecture to utilize friendship relationships in social networks for identity management, authorization and delegation. These policies only rely on social network structures (e.g., the weighted distances between pairs of entities), and support neither aggregate reputation nor flexible composition of reputation and credentials in trust policies.

7. CONCLUSIONS & FUTURE WORK

During everyday decision making, humans assess the trustworthiness of others by combining the notions of vertical trust and horizontal trust. *Vertical trust* captures the trust relationships that exist between individuals and institutions, while *horizontal trust* represents the trust that can be inferred from the observations and opinions of others. Although both types of trust have been explored within the context of distributed computing systems, effectively composing policies that rely on both types of trust has not been explored. In this paper, we took the first steps towards designing a composite trust management framework for distributed systems. We began by exploring a number of motivating scenarios to derive set of functional requirements for composite trust management policy languages. We then defined the syntax and semantics of the CTM language, which extends the *RT* family of trust management languages to allow arbitrary composition of horizontal and vertical trust metrics. We showed through a series of examples that CTM satisfies each of the functional requirements that our use case analysis uncovered, and that CTM can indeed express a wide-range of interesting policies.

In the future, we plan to develop a comprehensive trust management framework based upon CTM and explore the systems issues that emerge when trying to efficiently check compliance with CTM policies. Minimizing the computational overheads associated with policy evaluation is critical to the long-term success of this research effort. To this end, we plan to design and evaluate algorithms for finding (near) optimal execution plans for checking compliance with CTM policies. We also plan on investigating metrics for estimating the uncertainty introduced into the composite trust management process by factors such as end-host unavailability and credentials/feedback reports that are protected by layers of release policies that cannot be satisfied by a particular policy evaluator.

8. REFERENCES

[1] Amazon.com: Recommended for you. Web Site, Dec. 2008. http://www.amazon.com/gp/yourstore/recs/.

[2] S. Axelsson. Intrusion detection systems: A survey and taxonomy. Dept. of Computer Engineering Technical Report 99-15, Chalmers University of Technology, Mar. 2000.

[3] L. Bauer, S. Garriss, and M. K. Reiter. Distributed proving in access-control systems. In *Proceedings of the 2005 IEEE Symposium on Security and Privacy*, pages 81–95, May 2005.

[4] M. Y. Becker and P. Sewell. Cassandra: Distributed access control policies with tunable expressiveness. In *Proceedings of the Fifth IEEE International Workshop on Policies for Distributed Systems and Networks*, pages 159–168, June 2004.

[5] E. Bertino, E. Ferrari, and A. C. Squicciarini. Trust-X: A peer-to-peer framework for trust establishment. *IEEE Transactions on Knowledge and Data Engineering*, 16(7):827–842, July 2004.

[6] B. K. Bhargava and Y. Zhong. Authorization based on evidence and trust. In *International Conference on Data Warehousing and Knowledge Discovery*, pages 94–103, Aix-en-Provence, France, Sept. 2002.

[7] J. Biskup and Y. Karabulut. A hybrid pki model: Application to secure mediation. In *DBSec*, pages 271–282, 2002.

[8] M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. In *IEEE Symposium on Security and Privacy*, Oakland, CA, May 1996.

[9] M. Blaze, J. Feigenbaum, and M. Strauss. Compliance checking in the PolicyMaker trust management system. In *Proceedings of the Second International Conference on Financial Cryptography*, number 1465 in Lecture Notes in Computer Science, pages 254–274. Springer, Feb. 1998.

[10] P. Bonatti, C. Duma, D. Olmedilla, and N. Shahmehri. An integration of reputation-based and policy-based trust management. In *Sematic Web and Policy Workshop*, Galway, Ireland, Nov. 2005.

[11] B. Carminati, E. Ferrari, and A. Perego. Enforcing access control in web-based social networks. *ACM Transactions in Information and System Security*. to appear.

[12] B. Carminati, E. Ferrari, and A. Perego. Combining social networks and semantic web technologies for personalizing web access. In *Proceedings of the 4th International Conference on Collaborative Computing: Networking, Applications and Worksharing (CollaborateCom)*, Nov. 2008.

[13] B. Carminati, E. Ferrari, and A. Perego. A decentralized security framework for web-based social networks. *International Journal of Information Security and Privacy*, 2(4):22–53, 2008.

[14] F. Cornelli, E. Damiani, S. D. C. di Vimercati, S. Paraboschi, and P. Samarati. Choosing reputable servents in a p2p network. In *Proceedings of the 11th international conference on World Wide Web*, pages 376–386, 2002.

[15] E. Damiani, D. C. di Vimercati, S. Paraboschi, P. Samarati, and F. Violante. A reputation-based approach for choosing reliable resources in peer-to-peer networks. In *Proceedings of the 9th ACM conference on Computer and communications security*, pages 207–216, 2002.

[16] K. Frikken, M. Atallah, and J. Li. Attribute-based access control with hidden policies and hidden credentials. *IEEE Transactions on Computers*, 55(10):1259–1270, 2006.

[17] M. Grimsley, A. Meehan, G. Green, and B. Stafford. Social capital, community trust, and e-government services. In *International Conference on Trust Management*, Pisa, Italy, May 2003.

[18] K. Hoffman, D. Zage, and C. Nita-Rotaru. A survey of attack and defense techniques for reputation systems. *ACM Computing Surveys*, to appear.

[19] T. Jim. SD3: A trust management system with certified evaluation. In *Proceedings of the IEEE Symposium on Security and Privacy*, pages 106–115, May 2001.

[20] A. Josang, R. Ismail, and C. Boyd. A survey of trust and reputation systems for online service provision. *Decis. Support Syst.*, 43(2):618–644, 2007.

[21] S. Kamvar, M. Schlosser, and H. Garcia-Molina. EigenRep: Reputation Management in P2P Networks. In *Twelfth International World Wide Web Conference*, 2003.

[22] S. Kruk, S. Grzonkowski, A. Gzella, T. Woroniecki, and H.-C. Choi. D-foaf: Distributed identity management with access rights delegation. In *Asian Semantic Web Conference*, Beijing, China, Sept. 2006.

[23] A. J. Lee and M. Winslett. Towards an efficient and language-agnostic compliance checker for trust negotiation systems. In *Proceedings of the 3rd ACM Symposium on Information, Computer and Communications Security (ASIACCS 2008)*, pages 228–239, Mar. 2008.

[24] N. Li and J. C. Mitchell. RT: A role-based trust-management framework. In *Proceedings of the Third DARPA Information Survivability Conference and Exposition (DISCEX III)*, pages 201–212, Apr. 2003.

[25] N. Li, J. C. Mitchell, and W. H. Winsborough. Beyond proof-of-compliance: security analysis in trust management. *Journal of the ACM*, 52(3):474–514, 2005.

[26] N. Li, W. H. Winsborough, and J. C. Mitchell. Distributed credential chain discovery in trust management. *Journal of Computer Security*, 11(1):35–86, 2003.

[27] Liberty alliance project. Web Site, Dec. 2008. `http://www.projectliberty.org/`.

[28] A. Mounji, B. L. Charlier, D. Zampunieris, and N. Habra. Distributed audit trail analysis. In *Proceedings of the 1995 Symposium on Network and Distributed System Security (SNDSS'95)*, 1995.

[29] NetFlix prize: Home. Web Site, Dec. 2008. `http://www.netflixprize.com/`.

[30] R. Sandhu, E. J. Coyne, H. L. Feinstein, and C. E. Youman. Role-based access control models. *IEEE Computer*, 29(2):38–47, Feb. 1996.

[31] K. E. Seamons, M. Winslett, T. Yu, B. Smith, E. Child, J. Jacobson, H. Mills, and L. Yu. Requirements for policy languages for trust negotiation. In *IEEE International Workshop on Policies for Distributed Systems and Networks (POLICY)*, June 2002.

[32] Shibboleth Project. `http://shibboleth.internet2.edu/`.

[33] TrustBuilder2 download page. Web site, Oct. 2008. `http://dais.cs.uiuc.edu/dais/security/tb2/`.

[34] L. Wang, D. Wijesekera, and S. Jajodia. A logic-based framework for attribute based access control. In *Proceedings of the Second ACM Workshop on Formal Methods in Security Engineering (FMSE 2004)*, pages 45–55, Oct. 2004.

[35] D. Wijesekera and S. Jajodia. A propositional policy algebra for access control. *ACM Transactions on Information and Systems Security (TISSEC)*, 6(2):286–325, May 2003.

[36] W. H. Winsborough, K. E. Seamons, and V. E. Jones. Automated trust negotiation. In *DARPA Information Survivability Conference and Exposition*, Jan. 2000.

[37] L. Xiong and L. Liu. A reputation based trust model for peer-to-peer ecommerce communities. In *IEEE International Conference on E-Commerce (CEC)*, 2003.