QTM: Trust Management with Quantified Stochastic Attributes

NYU Computer Science Technical Report TR2003-848

Eric Freudenthal and Vijay Karamcheti
Courant Institute of Mathematical Sciences
New York University
{*freudent,vijayk*}*@cs.nyu.edu*

Abstract

Trust management systems enable the construction of access-control infrastructures suitable for protecting sensitive resources from access by unauthorized agents. The "state of the art"in such systems (i) provide fail-safe in that access will be denied when authorizing credentials are revoked, (ii) can mitigate the risk of insider attacks using mechanisms for threshold authorization in which several independent partially trusted agents are required to co-sponsor sensitive activities, and (iii) are capable of enforcing intra- and inter-organizational access control policies.

Despite these advantages, trust management systems are limited in their ability to express partial trust. Additionally, they are cumbersome to administer when there are a large number of related access rights with differing trust (and thereby access) levels due to the need for explicit enumeration of the exponential number of agent combinations. More importantly, these systems have no provision for fault tolerance in cases where a primary authorization is lost (perhaps due to revocation), but others are available. Such situations may result in a cascading loss of access and possible interruption of service.

In this pape, we propose extending traditional trust management systems through a framework of reliability and confidence metrics. This framework naturally captures partial trust relationships, thereby reducing administrative complexity of access control systems with multiple related trust levels and increasing system availability in the presence of authorization faults while still maintaining equivalent safety properties.

1 Introduction

Trust management (TM)[1, 6, 4, 5, 3, 2, 8] systems are application-independent infrastructures that can be used to enforce access control policies both within and between organizations.[1]

Throughout this paper, we will describe TM, and our stochastically-quantified trust management alternative in the context of access-control enforcement for a fictional bank named "Lou's Loans," a sole proprietorship owned by a gentleman named "Lou Learner." Lou's security policies express the authorization of tellers, managers, and other employees to perform sensitive operations such as endorsing transactions.

TM systems map access rights into abstract classes whose permissions can be delegated. Through delegation, these abstract classes can be organized into a hierarchy that mirrors the structure of an organization, including support for appropriate local autonomy in policy administration, thereby providing substantial advantages over access control lists (ACLs).

For example, the right to administer the rights available to employees a particular corporate unit can be directly administered by members of the organizational entity with appropriate jurisdiction. The same

[1]We classify role-based access control systems as examples of trust management systems.

mechanisms can be used to extend trust in coalition settings to a collaborating unit in another organization. In such cases, the right to administer the external group can also be delegated to agents in the external organization, providing a seamless chain of trust among eligible agents.

Despite these features, a notable shortcoming in current-day TM systems is that policies can only represent the boolean delegation (or, when not delegated, denial) of complete trust to act in some role. Therefore a unique role must be explicitly defined and administered for every "equivalence" class of "trusted" agent. Doing so results in significant increase in administrative complexity, and a reduction of the "quality" of authorization provided to an organization as described below:

- When multiple endorsement is required to achieve higher levels of aggregate trust (e.g. a requirement in our example application that multiple bank tellers co-endorse large cash transactions), TM systems require that all authorized combinations of distinct roles be enumerated.[2]

  When there are many roles representing related but different levels of trust, the resulting rule-set becomes large, complex, and arcane. Attempts to simplify such rule-sets result in approximations that either (1) deny access to combinations of agents that should be authorized or (2) create security risks due to inappropriate authorization of inappropriate agent combinations.

- TM does not provide a convenient mechanism to represent agents with equivalent individual roles but lesser aggregate trust. For example, consider three bank tellers A, B, and C, all of the same "rank". If A and B are married, then a lower level of aggregate trust should probably be conferred to transactions co-endorsed by A and B than to co-endorsed transactions that include C. Expression of each such coupling of trust between agents significantly increases the complexity of the set of policies that represent eligible co-authorization.

- While TM systems provide a natural representation of organizational structure, there is no mechanism to represent the transitive degradation of authority (and therefore organizational trust) of agents who are several steps removed from central administration other than through the definition of (still) more locally administered roles and complex policies that define their interactions.[3]

- While a TM system can be defined to implement arbitrary access control policies in an organization, no mechanism is provided to directly assess whether a system is consistent with an analytically sound risk analysis. As a result, the consistency and completeness of such systems are difficult to audit.

- In a system where credentials can be revoked, the cascading loss of authorization caused by the revocation of an administrative credential will leave an organization temporarily unable to conduct essential operations. An analysis of "weak link" credentials can assist in the restructuring of administrative policy to reduce the probability of such events. However, no representation of the probability of credential revocation, nor a framework for evaluating their impacts is included in conventional (boolean) TM systems.

Our recent dRBAC[3], and Li and Winsboro's RT[5] systems address some of these shortcoming by providing mechanisms to numerically modulate access rights within an abstract role. However, even these systems do not provide mechanisms useful to increase the scalability or the expressiveness of the encoding of co-endorsed authorization.

In this paper, we present an extension to conventional TM that incorporates trust metrics into authorization decisions. These metrics, are encoded as probabilities that an agent's actions will conform to an

[2] K-of-m authorization provides an alternative mechanism to enumerate sets of authorized agents.

[3] Some TM systems include mechanisms to limit transitive delegation to either a fixed level of indirections *independent* of the reliability of the issuers of such delegations or to disallow further delegation. These simple mechanisms do not incorporate partial trust or reliability metrics into delegation decisions.

organization's objectives. We describe a framework for representing and reasoning about the degradation of transitive trust, aggregation of redundant trust due to multiple endorsement, and shared trust among coupled parties.

The remainder of this paper is organized as follows: Section 2 describes the basic mechanisms of conventional "boolean" trust management systems and illustrates their limitations when used to enforce access-control policies in complex organizations. The following section (3) presents our quantified trust management (QTM) framework and the computation of composite trust in systems that include transitive, aggregate, and administrative delegations and work through a detailed example highlighting the main concepts.. Finally, approaches to constructing automatic evaluators of composite trust are described in Section 4. We conclude with a discussion of open research questions and future work in section 5

2 Background

In this paper, we use the term "trust management" (TM) to include both traditional trust management and role-based access control systems that represent equivalence classes of agents "trusted" to perform classes of restricted activities denoted by named abstract *roles*. We loosely define "restricted operations" as actions that require authorization, and "agents" as identified human or automated entities capable of endorsing requests for protected transactions or authoring credentials.[4]

Security policies associated with protected operations in TM systems are represented by associating each protected operation with a corresponding role. Access decisions in a TM framework involve the determination of whether the set of agents requesting a protected operation have been delegated "trust" to function in the role associated with the requested action.

2.1 Delegations

Role-based access control and trust management systems express a transitive trust relationship (conferring access rights) using statements called *delegations*. TM delegations are monotonic in that delegations only provide evidence supporting authorization, and the invalidation or omission of a delegation cannot result in greater access than if the "lost" delegation was available.

While a variety of delegation syntax and semantics have been proposed with the ability to express similar relationships, they more-or-less capture the same concept. In this paper, we present delegations using the syntax of dRBAC[3], where a delegation has the form:

$$\{subject \rightarrow object\}issuer$$

where *subject* identifies either a particular agent or a role representing an equivalence class of agents, the right-arrow ($\rightarrow$) represents a delegation of permissions, and *object* role identifies an access permission, and *issuer* identifies the agent that wrote the delegation. Such a delegation would indicate that an agent with rights associated with the *subject* is entitled to access rights associated with the *object*. provided that (1) the delegation is authentic,[5] and (2) the issuer is authorized to control access to the rights represented by object.

We note that a delegation's object both represents an equivalence class of permissions, and an equivalence class of agents granted access to those permissions. The generalization of these equivalence classes to a single abstraction called a *role* allows for delegations to be transitively chained when a particular role R appears as both the object of one delegation (representing a R_P, the set of permissions represented by R) and the subject of another (representing R_A, the set of agents with permissions represented by R).

[4]The authoring of credentials can also be viewed as a protected operation.

[5]In distributed systems, this verification is implemented using standard cryptographic techniques.

$$\{\ \text{L.manager} \rightarrow \text{L.SrTeller'}\ \}\ \text{L} \quad (1)$$
$$\{\ \text{Max} \rightarrow \text{L.manager}\ \}\ \text{L} \quad (2)$$
$$\{\ \text{Tom} \rightarrow \text{SrTeller}\ \}\ \text{Max} \quad (3)$$

Figure 1: Authorization of Tom to have the privileges of a Senior Teller

$$\{\ \text{L.tellerTrainee} \rightarrow \text{L.enterOffice}\ \}\ \text{L} \quad (4)$$
$$\{\ \text{L.tellerTrainee} \rightarrow \text{L.endorseWithdrawLow}\ \}\ \text{L} \quad (5)$$
$$\{\ \text{L.tellerJr} \rightarrow \text{L.tellerTrainee}\ \}\ \text{L} \quad (6)$$
$$\{\ \text{L.tellerJr} \rightarrow \text{L.endorseWithdrawalMed}\ \}\ \text{L} \quad (7)$$
$$\{\ \text{L.tellerSr} \rightarrow \text{L.tellerJr}\ \}\ \text{L} \quad (8)$$
$$\{\ \text{L.tellerSr} \rightarrow \text{L.modifyAccount}\ \}\ \text{L} \quad (9)$$

Figure 2: Basic Authorization of Tellers in a Trust Management Framework

Extending the SDSI/SPKI[8] model, access rights (defined as roles) are defined within an agent's namespace,[6] and the owner of a namespace does not require additional authorization to delegate them. Authorization to delegate a role in another's namespace is represented by a corresponding *administrative role* denoted by a tick (') mark which can also be delegated to arbitrary agents through delegation credentials.

In our example, all roles representing an access right are defined within Lou's namesapce (abbreviated as "L."). The three delegations in Figure 1 authorize Tom to act in the role of a senior teller in Lou's bank as follows:

- In delegation (1), Lou delegates trust to administer the teller role defined in his namespace to managers, also defined in his namespace.
- In delegation (2), Lou delegates trust to act in the manager role defined in his namespace to Max.
- In delegation (3), Max delegates trust to act in the senior teller role as defined in Lou's namespace to Tom.

Due to the monotonicity of trust in boolean TM, an orderable set of roles can be compactly represented. For example, the delegations in Figure 2 define a hierarchical relationship between tellers of multiple ranks using an idiom that inverts the natural relationship between ranks, where subordinates are given some portion of the authorization available to their superiors. However, this compact idiom of role inversion within an organizational hierarchy is not suitable when access rights are not monotonically orderable.

In order to limit his exposure to "insider risk" due to mistakes or fraud, Lou may only want to extend limited trust to individual tellers, perhaps only authorizing individual tellers to endorse withdrawals of limited value, and requiring multiple tellers to endorse larger transactions. Furthermore, some tellers that have worked for several years for the bank may be trusted to a greater degree than those recently hired.

[6] In distributed systems, agents (and thereby their namespaces) are commonly associated with public keys.

Singly-endorsed Withdrawals

$$\{\text{ L.tellerTrainee} \rightarrow \text{L.endorseWithdrawLow } \}\text{ L} \quad (10)$$

$$\{\text{ L.tellerJr} \rightarrow \text{L.endorseWithdrawMed } \}\text{ L} \quad (11)$$

Co-endorsed Withdrawals

$$\{\text{ L.tellerSr, L.tellerSr} \rightarrow \text{L.endorseWithdrawHigh } \}\text{ L} \quad (12)$$

$$\{\text{ L.tellerJr, L.tellerJr, L.tellerSr} \rightarrow \text{L.endorseWithdrawHigh } \}\text{ L} \quad (13)$$

$$\{\text{ L.tellerJr, L.tellerJr, L.tellerJr , L.tellerJr} \rightarrow \text{L.endorseWithdrawHigh } \}\text{ L} \quad (14)$$

Figure 3: Authorization of Tellers to Endorse Withdrawals in a boolean TM Framework

TM systems require explicit enumeration of all intermediate teller trust levels (e.g. tellerTrainee, tellerJr, tellerSr) and authorized co-endorsement combinations. A variety of mechanisms have been proposed to encode authorized co-endorsement combinations. A common mechanism enumerates potential co-endorser's roles as a delegation with multiple subjects as is illustrated in Figure 3. In this framework, large transactions require authorization of two senior tellers or greater numbers of more junior tellers.

As this example shows, the number of delegations required for a complex organization can increase exponentially with the number of roles representing distinct trust levels and access rights, resulting in a morass of delegations that is difficult to construct and analyze for correctness.

We also identify three additional challenges endemic to boolean TM frameworks:

- Systems of boolean trust attributes are difficult to extend or modify: the introduction of a new role with distinct trust levels or a modification of the level-of-trust associated with a role may require the creation or replacement of many delegations.
- No convenient framework is available to represent organizationally known unusual circumstances that reduce trust. For example, while Lou may not want to permit co-endorsement of large withdrawals by a pair of senior tellers who are married to each other, encoding appropriate trust properties to this couple is an onerous administrative task in a boolean TM system.
- Boolean trust-management systems provide only crude mechanisms to limit the propagation of access rights through third-party delegations. Current delegation encodings provide either (1) an integer limit to the number of transitive links through which authorization can be transmitted, or (2) a boolean flag that, if set, disallows further transitive chaining.

2.2 Modulated Rights Within a Role

Recent work (e.g. our recent dRBAC[3], and Li and Winsboro's RT[5]) have included representations of roles with modulated attributes. RT_n provides a mechanism where each role's definition includes a vector of scalar values; In this system, delegations include the definition of a function that projects a subject vector into an object vector. dRBAC takes a different approach: Trust attenuation attributes are defined within namespaces, and a delegation can attenuate any of these attributes by direct reference. While these mechanisms can be used to define a family of access levels associated with each role, they do not provide mechanisms useful for increasing the scalability of co-endorsed authorization.

3 QTM: Quantified Trust Management

In this section, we describe a unifying extension to trust management frameworks that directly expresses partial trust in delegations. This direct encoding of partial trust enables the construction of a trust management system that can analytically determine the "composite" trust provided to an agent. This computed "trust level" can then be used to determine the level of access that should be provided to requesters.

QTM expresses partial trust as reliability metrics that provide a foundation for a monotonic trust management framework which directly expresses partial trust relationships and facilitates the construction of access control systems to enforce complex organizational policies.

3.1 Stochastic Attributes

QTM reliability metrics are encoded as attributes that represent stochastic boolean processes. This stochastic abstraction facilitates the use of standard statistical frameworks to analyze the effective "composite" trust available to agents requesting to perform sensitive operations by a set of delegations.

QTM expresses the reliability of particular agent characteristics that are relevant for access control decisions for agents acting in some object role as the expected value (a probability in the range $[0..1]$) of a boolean function on random boolean variables. The semantics of these characteristics are not defined by QTM, thus making QTM suitable for a variety of applications.

We define two types of attribute bindings in delegations. "Singleton bindings" associate a single random variable with each delegation, where "agent bindings" associate a distinct random variable with each agent being delegated the object role. These attributes bindings are encoded within delegations as comma-separated attribute binding lists as follows:

$$\{\ \text{subject} \rightarrow \text{object}\ (\ \text{attribute\_binding\_list}\ )\ \}\ \text{issuer} \tag{15}$$

A singleton attribute binding has the form:

$$\text{attribute}* = [\textit{value}]$$

and an agent-duplicated attribute binding has the form

$$\text{attribute}* = [\textit{value}]a$$

where *value* is a probability between 0 and 1. Attributes not enumerated within an attribute binding list are implicitly bound to a random boolean variable with an expected value of one.

Given a set of QTM delegations D, we represent we define the composite reliability an agent γ acting in some object ω role in the context of some attribute α as the expected value of a boolean function $V^{\alpha}_{\gamma \Rightarrow \omega}$ that references random variables corresponding to attribute bindings within authorizing delegations. To assist in constructing proofs, we also define a parallel form $V^{\alpha}_{\sigma \Rightarrow \omega,\gamma}$ where γ is acting in a subject role σ. Finally, when multiple agents G are all acting in role ω, their composite reliability is the expected value of $\bigcup_{\gamma \in G} V^{\alpha}_{\gamma \Rightarrow \omega}$

Below we describe how to compute V.

When reasoning over the aggregate reliability of a system of delegations that express partial reliability, we name random variables within a set of delegations shown below:

- v^{α}_{δ} is the singleton random variable associated with attribute α in delegation δ.
- $v^{\alpha}_{\delta,\gamma}$ is the agent-duplicated random variable associated with attribute α and agent γ in delegation δ.

3.2 Reliability Conferred by a Delegation

For any delegation δ issued by an agent i with object ω, its reliability in the context of α when authorizing agent γ is given by

$$v^{\alpha}_{\delta,\gamma} \wedge V^{\alpha}_{i \Rightarrow \omega'}$$

where ω' is ω's administrative role.

As the root of all trust chains, agents are attributed full reliability for all roles within their namespaces. Therefore, for all attributes α, if object role ω is defined within γ's namespace, we define $V^{\alpha}_{\gamma \Rightarrow \omega'} = 1$.

3.2.1 Example: Direct Conferring of Trust

In the context of his bank, Lou may require that customer service workers that approve medium sized cash transactions be *expected* to endorse transactions in "good faith" with a probability of at least 0.995. Lou could associate this probability of a customer service agent acting with integrity when handling high-value transactions with a named stochastic attribute defined in his namespace "L.integ," by issuing the following delegations providing Cal, Curt, and Chris "trust" (and thereby access) commensurate with his appraisals of their character.

$$\{ \text{Cal} \rightarrow \text{L.cserv (L.integ*=0.997 } \} \text{ L} \tag{16}$$

$$\{ \text{Curt} \rightarrow \text{L.cserv (L.integ*=0.997) } \} \text{ L} \tag{17}$$

$$\{ \text{Chris} \rightarrow \text{L.cserv (L.integ*=0.99) } \} \text{ L} \tag{18}$$

Applying the rules describe above, computing Cal's reliability in role L.cserv as the expected value $E(V^{L.integ}_{Cal \Rightarrow L.cserv})$ is straightforward. First, by the definition of V:

$$V^{L.integ}_{Cal \Rightarrow L.cserv} = v^{L.integ}_{16} \wedge V^{L.integ}_{L \Rightarrow L.cserv'}$$

and

$$V^{L.integ}_{L \Rightarrow L.cserv'} = 1.$$

Therefore,

$$V^{L.integ}_{Cal \Rightarrow L.cserv} = v^{L.integ}_{16}$$

and

$$E(V^{L.integ}_{Cal \Rightarrow L.cserv}) = 0.997.$$

By applying similar reductions, it can be easily demonstrated that $E(V^{L.integ}_{Curt \Rightarrow L.cserv}) = 0.997$ and $E(V^{L.integ}_{Chris \Rightarrow L.cserv}) = 0.99$, indicating that Cal and Curt have sufficient reliability as customer service agents, and that Curt does not.

3.2.2 Example: Propagation of Administrative Reliability

Let's assume that Lou hires "Max" as a manager and empowers him to authorize customer service agents with the following delegation

$$\{\text{Max} \rightarrow \text{L.cserv' (L.integ*=0.999) } \} \text{ L.} \tag{19}$$

Note that delegation 19 indicates that Lou assesses Max's integrity as 0.999 when acting as an administrator of the role L.cserv. Since Lou is the owner of the L.cserv namespace (and therefore $E(V^{L.integ}_{Lou \Rightarrow L.cserv'}) = 1$),

$$E(V^{L.integ}_{Max \Rightarrow L.cserv'}) = 0.999.$$

Should Max now hire Tom and Tim, whom he asserts integrity reliabilities of 0.97 and 0.998, respectively, he might issue the following delegations:

$$\{ \text{Tom} \rightarrow \text{L.cserv (L.integ*=0.97)} \} \text{ Max} \quad (20)$$

$$\{ \text{Tim} \rightarrow \text{L.cserv (L.integ*=0.998)} \} \text{ Max} \quad (21)$$

Tom and Tim's expected reliability can be computed by applying the rules presented above,

$$V^{L.integ}_{Tom \Rightarrow L.cserv} = v^{L.integ}_{20} \wedge V^{L.integ}_{Max \Rightarrow L.cserv'} \quad (22)$$

$$v^{L.integ}_{20} \wedge v^{L.integ}_{19} \quad (23)$$

$$E(V^{L.integ}_{Tom \Rightarrow L.cserv}) \approx 0.97 \quad (24)$$

$$(25)$$

$$V^{L.integ}_{Tim \Rightarrow L.cserv} = v^{L.integ}_{21} \wedge V^{L.integ}_{Max \Rightarrow L.cserv'} \quad (26)$$

$$v^{L.integ}_{21} \wedge v^{L.integ}_{19} \quad (27)$$

$$E(V^{L.integ}_{Tim \Rightarrow L.cserv}) \approx 0.997 \quad (28)$$

Note that the attenuation of Tim's expected integrity is, as is appropriate, a direct and rational extension of the partial trust that Lou placed in Max.

3.3 Composition of Reliability through Transitive and Parallel Authorization

Trust is attenuated when transitively conferred through unreliable delegations. As with a communication channel transmitted over several links, each with independent reliability characteristics, QTM evaluates a transitively linked chain of delegations in terms of the joint reliability of the delegations that compose the chain.

In contrast, co-endorsement by multiple parties can provide higher reliability. As with delegated and conferred transitive trust, QTM, the trust conferred through multiple authorization paths corresponds directly to the ontology being expressed. In this context QTM abstracts each a chain of delegations as a partially reliable channel, and if an event is reported by all channels, then the probability that the event did *not* occur is the is joint probability of all channels failing. More formally, if P is a set of redundant authorization paths, each with their own expected reliability, expressed as a stochastic process, then the aggregate reliability is the expected value of the union of the stochastic processes representing the reliabilities of each path in P.

3.3.1 Example of Transitive and Parallel Delegation

Extending our example, consider Lou's challenge in authorizing a third teller named Trish, who is married to Tim. Recall that, if Tim and Trish are a married couple, then it would be prudent for Lou to not consider their integrity independent since they have a higher risk of joint conspiratorial behavior than two more independent employees. In order to limit this "insider risk", while still conferring to them appropriate *aggregate* trust when they endorse individually or with other custimer service agents, Lou defines a new role "L.timTrish" representing the trust conferred to their family (in delegation 29). In this delegation, the trust afforded to their family is represented by a single stochastic process with probability 0.999. He then invalidates delegation 21 and issues delegations 30 and 31 (below):

$$\{ \text{L.timTrish} \rightarrow \text{L.cserv (L.integ*=0.999)} \} \text{ L} \quad (29)$$

$$\{ \text{Tim} \rightarrow \text{L.timTrish (L.integ*=0.999)} \} \text{ L} \quad (30)$$

$$\{ \text{Trish} \rightarrow \text{L.timTrish (L.integ*=0.999)} \} \text{ L} \quad (31)$$

Note that this direct encoding of the correlated reliability within their family affords Tom and Trish, when acting alone or in collaboration with others, the full trust that would be available if they were not married (0.998) and collaboration within their family is properly evaluated as more reliable than individual action (0.9985), but less reliable than collaboration with non-family members.

Note that, in contrast to the challenge of integrating a new correlated trust relationship into a conventional (boolean) TM system, this QTM representation of correlated trust only required modifications to the authorization provided to Tom and Trish.

3.4 Monotonicity of Authorization

Recall that the removal of credentials in that monotonic trust management systems never increases authorization, and therefore provides an important safety property on systems where credentials may be omitted or lost. QTM is a monotonic system, and the insertion of credentials beyond a base set sufficient to satisfy an authorization requirement can result in an increase in attributed reliability and provide redundancy should other delegation credentials become revoked or invalidated.

3.5 Extended Example: Computation of Transitive and Parallel Trust

Since they represent significant risk, Lou would like to require very high reliability (say, > 0.99999) for authorization of large transactions. While it is difficult to find employees with sufficient reliability, it is not difficult to hire multiple employees who, when co-endorsing a transaction, have sufficient aggregate authority.

He also would like to extend this reliability throughout the entire authorization chain, so that he also limits his risk of harm through malfeasance throughout his organization. He can achieve this by limiting the reliability that can be conferred to each employee through co-endorsement by limiting the authorization of delegated administrative roles to roles with limited authorization.

For example, consider the following system of delegations:

$$\{\ \text{L.teller} \rightarrow \text{L.cserv (L.integ*=0.999a)}\ \}\ \text{L} \tag{32}$$

$$\{\ \text{L.mgr} \rightarrow \text{L.teller' (L.integ*=0.999a)}\ \}\ \text{L} \tag{33}$$

$$\{\ \text{Max} \rightarrow \text{L.mgr}\ \}\ \text{L} \tag{34}$$

$$\{\ \text{Mel} \rightarrow \text{L.mgr}\ \}\ \text{L} \tag{35}$$

$$\{\ \text{Tom} \rightarrow \text{L.teller}\ \}\ \text{Max} \tag{36}$$

$$\{\ \text{Tom} \rightarrow \text{L.teller}\ \}\ \text{Mel} \tag{37}$$

$$\{\ \text{Tim} \rightarrow \text{L.teller}\ \}\ \text{Max} \tag{38}$$

$$\{\ \text{Tim} \rightarrow \text{L.teller}\ \}\ \text{Mel} \tag{39}$$

Delegations 32 and 33 define new roles for tellers and managers. These roles limit the reliability that can be attributed to any agent in these roles through aggregation to 0.999. Since the attributes bound in these two delegations specify agent-binding, an agent's reliability, when conferred through this delegation, is limited by an independent stochastic process, allowing several of them to aggregate their reliability through co-endorsement.

In order to directly represent withdrawal authorization Lou defines a new role "L.wd" that represents a withdrawal; the following delegation permits customer service workers to authorize such transactions:

$$\{\ \text{L.cserv} \rightarrow \text{L.wd'}\ \}\ \text{L} \tag{40}$$

This enables withdrawals to also be directly represented as virtual agents. We shall assert that WD1 is a large withdrawal that Lou's access controller determines requires authorization with expected integrity of at lest 0.99999 in the role of L.wd.

Tom and Tim issue the following delegations:

$$\{\text{ WD1} \rightarrow \text{L.wd }\}\text{ Tom} \tag{41}$$

$$\{\text{ WD1} \rightarrow \text{L.wd }\}\text{ Tim} \tag{42}$$

To determine if WD1 is authorized, the access controller now must evaluate $E(V^{L.integ}_{WD1 \Rightarrow L.wd})$ by first recursively expanding all random functions in $V^{L.integ}_{WD1 \Rightarrow L.wd}$, and then evaluating the resulting inter-dependent set of equations. First, the expansion:

$$V^{L.integ}_{WD1 \Rightarrow L.wd} = (v^{L.integ}_{41} \wedge V^{L.integ}_{Tom \Rightarrow L.wd', Tom}) \vee (v^{L.integ}_{42} \wedge V^{L.integ}_{Tim \Rightarrow L.wd', Tim}) \tag{43}$$

$$V^{L.integ}_{Tom \Rightarrow L.wd', Tom} = V^{L.integ}_{Tom \Rightarrow L.teller} \wedge V^{L.integ}_{L.teller \Rightarrow L.cservTom} \wedge V^{L.integ}_{L.cserv \Rightarrow L.wd'} \tag{44}$$

$$V^{L.integ}_{Tim \Rightarrow L.wd', Tim} = V^{L.integ}_{Tim \Rightarrow L.teller} \wedge V^{L.integ}_{L.teller \Rightarrow L.cserv, Tim} \wedge V^{L.integ}_{L.cserv \Rightarrow L.wd'} \tag{45}$$

$$V^{L.integ}_{Tom \Rightarrow L.teller} = ((v^{L.integ}_{36} \wedge V^{L.integ}_{Max \Rightarrow L.teller'}) \vee (v^{L.integ}_{37} \wedge V^{L.integ}_{Mel \Rightarrow L.teller'})) \tag{46}$$

$$V^{L.integ}_{Tim \Rightarrow L.teller} = ((v^{L.integ}_{38} \wedge V^{L.integ}_{Max \Rightarrow L.teller'}) \vee (v^{L.integ}_{39} \wedge V^{L.integ}_{Mel \Rightarrow L.teller'})) \tag{47}$$

$$V^{L.integ}_{Mel \Rightarrow L.teller'} = V^{L.integ}_{Mel \Rightarrow L.mgr} \wedge V^{L.integ}_{L.mgr \Rightarrow L.teller', Mel} \tag{48}$$

$$V^{L.integ}_{Max \Rightarrow L.teller'} = V^{L.integ}_{Max \Rightarrow L.mgr} \wedge V^{L.integ}_{L.mgr \Rightarrow L.teller', Mel} \tag{49}$$

$$V^{L.integ}_{Max \Rightarrow L.mgr} = v^{L.integ}_{34} \wedge 1 \tag{50}$$

$$V^{L.integ}_{Mel \Rightarrow L.mgr} = v^{L.integ}_{35} \wedge 1 \tag{51}$$

$$V^{L.integ}_{L.mgr \Rightarrow L.teller', Max} = v^{L.integ}_{33, Max} \wedge 1 \tag{52}$$

$$V^{L.integ}_{L.mgr \Rightarrow L.teller', Mel} = v^{L.integ}_{33, Mel} \wedge 1 \tag{53}$$

$$V^{L.integ}_{L.teller \Rightarrow L.cservTom} = v^{L.integ}_{32, Tom} \wedge 1 \tag{54}$$

$$V^{L.integ}_{L.teller \Rightarrow L.cservTim} = v^{L.integ}_{32, Tim} \wedge 1 \tag{55}$$

$$V^{L.integ}_{L.cserv \Rightarrow L.wd'} = v^{L.integ}_{40} \wedge 1 \tag{56}$$

$$\tag{57}$$

Unlike the examples presented earlier, this expansion is not simple to evaluate by inspection. Approaches to evaluating complex systems of delegations are examined in the next section.

4 Evaluating of Reliability of Authorization in QTM

4.1 System Components Involved in Authorization

In general, when validating a request to perform a restricted operation, a provider of a trust-sensitive resource will provide to an authorization evaluator:

- A description of the policy P to be enforced. In role-based access control, this is typically either a single role, or a set of roles that the requesting agents must show membership of. In QTM, the policy is a set of roles, each with a minimum reliability attribute binding.
- Identities of agent(s) R endorsing (or requesting) the transaction. Note that in some contexts, a transaction may (depending on context) either be be endorsed by the agent that requested it, or by some authorized third party or parties (e.g. a bank teller or customs agent).

- Optional evidence E, in the form of a set of delegations provided by P that prove their satisfaction of R. In the absence of evidence, a credential discovery mechanism may be available collect necessary delegations. Such mechanisms are described in [3, 5].

The result of such an evaluation is a binary result indicating whether the access control policy (P) is satisfied for R.

We note that, sufficient evidence may be already known to the the evaluator through an previously established channel,[7] and therefore evidence may not need to be provided with each authorization request. In addition, in order to amortize the cost of evaluation between requests, an authorization evaluator may "memoize" computed results for reuse in future transactions. In such cases, the authorization evaluator should provide a mechanism to determine when evidence of authorization either expires or is invalidated.

While, as in other RBAC or TM systems, issuers of delegations must demonstrate authority to delegate object roles, there is not need for additional authorization of probability attributes since they monotonically reduce access rights.

4.2 Evaluating Composite Trust

A QTM evaluator synthesizes probability attribute values over a set of credentials authorizing a trust relationship by (1) recursively expanding all functions in the request authorization equation (of the form $V^{\alpha}_{R \Rightarrow P}$ until the resulting fully expanded stochastic satisfiability (SSAT) equation, Q, that contains references to random variables and no references to functions and then (2) determining if the expected value of Q satisfies the authorization threshold.

The reduction of a QTM problem to Q is a linear time transformation. However, as an instance of a bounded SSAT problem, computing the expected value of Q is non-trivial. We identify three approaches:

- The well-known "inclusion-exclusion" technique, provides an exact solution, but is often of exponential complexity and therefore does not provide a general solution.
- Custom heuristic approaches such as [7] that exploit specific structural elements of SSAT problems, often yielding solutions in polynomial time.
- Monte Carlo simulation techniques yield approximations whose accuracy increases with the square-root of the number of trials and therefore is only suitable for problems whose acceptance threshold is not near to unity.

Custom heuristic solutions to SSAT may provide useful solutions. Unlike general SSAT problems, the equations emitted by QTM reductions contain no negations. We are investigating whether this restricted form of SSAT yields computational advantages.

Monte Carlo techniques are commonly used to approximate the solutions to SSAT problems. These approaches typically proceed by computing the solution of the system of equations for a large number of draws of the random variables. The expected standard deviation on the error of such an approximation is typically $1/\sqrt{N}$ where N is the number of trials. We observe that, for the problems we are considering, acceptable error in (for example) integrity computation must be fairly low, probably around 10^{-5}, requiring approximately 10^{10} trials to reduce standard deviation to the same range of values.

Since no random variable is complimented in our SSAT reductions, our results are monotonic. For this reason, all trails in which all variables are true must satisfy the equation and therefore are uninteresting. We enumerate the rare trials where variables are false using Gibbs generators, thus substantially reducing the number of trials to be evaluated.

[7] In centralized implementations, all authorized delegations may already be stored with the evaluator.

These sets of trials where variable assignments are false can be used to convert the problem into a data-flow structure where each sub-expression yields a reduced set of potentially failed trials. Such an evaluator can terminate early if the number of failed trials exceeds the acceptance threshold.

Our initial timing experiments conducted using a Python interpretor system are encouraging: The generation of trial sets dominates computation time. However, by pre-computing and memoizing a sets of "interesting" events, our Python implementation performs a boolean operation on about 10^6 elements each second; we expect two to three additional orders-of-magnitude speedup when the sets are stored as sorted vectors and interpreted code is replaced with an optimized native implementation.

To limit the standard deviation of error in the evaluation of the extended example presented in Section 3.5 to less than 10^{-5}, then the monte-carlo evaluator should evaluate 10^{10} rounds. Since all random variables in our example are false with a rate of 10^{-4}, then each input set (representing a random variable) will contain approximately 10^6 "interesting events". The computation includes fifteen boolean operations, each with an expected execution time of 1s using our Python evaluator, resulting in an expected computation time of 1.5s. We view this as encouraging since it should execute in approximately 0.01s if native code is available.

4.3 Continuous Monitoring

It is often desirable to continuously authorize a sustained relationship, however most TM implementations only provide mechanisms to authorize transactions at particular moments in time. Our dRBAC system provides an extended interface that permits access controllers request automatic notification when an established authorizing trust relationship is lost.. As described in [3], this model can yield computational efficiency since monitoring of authorization is executed asynchronously when needed rather than being inserted into the critical path of every transaction. Upon notification of the invalidation of a credential, dRBAC automatically recomputes available authorization using the remaining credentials, only notifying trust-sensitive systems when authorization can no longer be maintained. In addition, mechanisms are provided to provide additional delegations to a dRBAC evaluator to be kept in reserve in advance of a revocation or expiration, allowing for a conscientious agent to continuously maintain authorization through the lifetime of several delegation sets. These techniques are applicable to authorization systems that implement QTM.

5 Future Work and Open Problems

We have identified several potential areas for future research including online algorithms to evaluate QTM problems, and algorithms to determine the reliability of authorization provided by QTM.

5.1 Implementation of a QTM Evaluator

We are presently in the process of implementing an online evaluator for QTM authorization. This work will involve mechanisms to automatically reduce QTM authorization request into stochastic satisfiability (SSAT) problems and the construction of systems to evaluate and approximate their expected values.

We are implementing a more efficient Monte Carlo evaluator, and are also investigating the complexity of analytic and heuristic approaches to solving for and approximating the expected value of the restricted SSAT problem generated by QTM evaluators.

5.2 Multiple Stochastic Attributes

We are investigating several extensions to QTM including mechanisms to evaluate authorization decisions that depend on multiple attributes. For example, it would be desirable to independently encode different components of teller reliability, for example, integrity, knowledge, and judgment can all be independently

measured, and an access controller can potentially determine acceptable limits based on transaction characteristics. It appears that computation of their expected value should be coupled since an employee not acting with integrity will not use his knowledge of corporate procedures to the company's benefit. However, it is not clear whether this coupling should be expressed in the trust management framework or instead in access controllers' transaction requirements.

Management authority in an organization may be independent of operational attributes such as knowledge of details of operating procedures. We are investigating the exploitation of multiple attributes into delegations that will allow a uniform infrastructure to represent both management and operational authority.

5.3 Stability Analysis of Authorization Schemas

The disabling of access through loss-of-authorization can paralyze systems whose availability is of critical importance. Such failures can be due to inherently unreliable authorization policies (for example, delegating a critical responsibility to a single employee in poor health) or deliberate sabotage by malevolent insiders with administrative responsibilities. By enabling more complete representation of authorization policy than boolean TM systems, QTM systems are (in general) more resilient against such failures. However, we believe that analytical infrastructure to assist in the detection of "weak links" in authorization chains can and should be developed.

While articulation points and "min cuts" can be easily detected in authorization graphs, these may not actually be the "weak links" in a composite authorization scheme where agents have varied "availability" and integrity. We suggest that, instead, stochastic attributes that represent various aspects of agent availability can be used to analyze a system of delegations to determine a system's potential "weak links" in authorization chains.

Such algorithms can be used to construct dependability analysis tools that can help engineers and managers anticipate authorization failures and permit pro-active adjustment. We currently are investigating algorithms to compute the "authorization risk" associated with each delegation or agent.

Complex stateful systems are commonly modeled as a a network of Markov processes. A QTM problem can also be mapped to such a framework, and evaluators such as Sanders' UltraSAN[9] may be useful in determining system stability characteristics.

6 Summary

QTM simplifies the specification of complex access control policy through the incorporation of stochastic reliability attributes within trust-management credentials. These reliability attributes describe stochastic processes that model the expected reliability of an agents' actions. This paper has presented a syntax for and identified issues in reasoning about "composite trust" in QTM systems.

We believe that the resulting *quantitatively modulated* trust management systems will permit the efficient expression and incremental revision of complex security policies that are impractical to implement using current state-of-the-art trust management systems. Since QTM credentials directly represent statistical reliability metrics that directly correspond to organizational risk, we expect that the resulting systems will be easier to audit for correctness and completeness than their traditional "boolean" trust management predecessors.

We anticipate that QTM will facilitate the deployment of insider-attack resistant access control policies that require multiple endorsement of sensitive operations. These systems can directly and compactly encode correlated trust (such as co-endorsement by family members or other identified correlated interest groups). These encodings can diminish the trust afforded to groups of agents with correlated trust, thereby requiring diversity in authorization. Conversely, by providing a direct mechanism to express partial trust relation-

ships, we anticipate that these systems will provide more ”complete” authorization, allowing many or all combinations of agents with sufficient aggregate trust to authorize sensitive operations.

Finally, we expect that proposed framework will enable the construction of tools that analyze authorization schemas for vulnerability to various denial-of-service failures due to agent availability metrics. These tools can potentially be used to guide the reinforcement of vulnerable systems through changes of policy and/or staffing.

Acknowledgements

This research was sponsored by DARPA agreements N66001-00-1-8920, and N66001-01-1-8929; and by NSF grants CAREER:CCR-9876128 and CCR-9988176. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any copyright annotation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as representing the official policies or endorsements, either expressed or implied, of DARPA, SPAWAR SYSCEN, or the U.S. Government.

References

[1] Matt Blaze, Joan Feigenbaum, and Jack Lacy. Decentralized trust management. In *Proc. of IEEE Conf. on Privacy and Security*, 1996.

[2] Carl M. Ellison, Bill Frantz, Butler Lampson, Ron Rivest, Brian M. Thomas, and Tatu Ylonen. SPKI Certificate Theory. IETF RFC 2693, 1998.

[3] Eric Freudenthal, Tracy Pesin, Lawrence Port, Edward Keenan, and Vijay Karamcheti. dRBAC: Distributed Role-Based Access Control for Dynamic Coalition Environments. In *Proceedings of the Twenty-second IEEE International Conference on Distributed Computing Systems (ICDCS), Vienna, Austria*, 2002.

[4] Ninghui Li, Benjamin N. Grosof, and Joan Feigenbaum. A practically implementable and tractable delegation logic. In *Proc. of IEEE Symp. on Security and Privacy*, 2000.

[5] Ninghui Li, John C. Mitchell, and William H. Winsborough. Design of a role-based trust management framework. In *Proceedings of the 2002 IEEE Symposium on Security and Privacy*, pages 114–130. IEEE Computer Society Press, May 2002.

[6] M. Winslett, T. Yu, K. E. Seamons, A. Hess, J. Jacobson, R. Jarvis, B. Smith, and L. Yu. Negotiating trust on the web. *IEEE Internet Computing*, 6(6):30–37, 2002.

[7] Michael L. Littman and Stephen M. Majercik and Toniann Pitassi. Stochastic boolean satisfiability. *Journal of Automated Reasoning*, 27(3):251–296, 2001.

[8] Ronald L. Rivest and Butler Lampson. SDSI – A simple distributed security infrastructure. In *Proc. of CRYPTO'96*, 1996.

[9] William H. Sanders and W. Douglas Obal (II) and Muhammad A Quershi and F. K. Widjarko. The ultrasan modeling environment. *Performance Evaluation*, 24(1-2):89–115, 1995.