CERIAS Tech Report 2005-100

ACCESS CONTROL MANAGEMENT IN A DISTRIBUTED ENVIRONMENT SUPPORTING DYNAMIC COLLABORATION

by Basit Shafiq , Elisa Bertino, and Arif Ghafoor

Center for Education and Research in
Information Assurance and Security,
Purdue University, West Lafayette, IN 47907-2086

Access Control Management in a Distributed Environment Supporting Dynamic Collaboration

Basit Shafiq
School of Electrical and Computer Engineering, Purdue University
West Lafayette, IN, USA
shafiq@ecn.purdue.edu

Elisa Bertino
Department of Computer Sciences and CERIAS, Purdue University
West Lafayette, IN, USA
bertino@cerias.purdue.edu

Arif Ghafoor
School of Electrical and Computer Engineering, Purdue University
West Lafayette, IN, USA
ghafoor@ecn.purdue.edu

ABSTRACT
Ensuring secure and authorized access to remote services and information resources in a dynamic collaborative environment is a challenging task. Two major issues that need to be addressed in this regard are: specification of access control requirements and trust management. Specification of access control requirements for dynamic collaboration is challenging mainly because of the limited or lack of knowledge about remote users' identities and affiliations. The access control policies and constraints defining users' authorization over remote resources and services need to be specified in terms of the attributes and properties of the users. Moreover, the criteria for validating the attributes of the users should also be specified as part of access control requirements. Trust management, in the context of dynamic collaboration, involves validation of user's attributes for secure interaction and prevention of unauthorized disclosure of policies and attributes. The paper discusses these issues in detail and presents a framework for access control and trust management in a distributed collaborative environment.

Categories and Subject Descriptors
D.4.6 [**Security and Protection**]: Access Controls; H.2.7 [**Database Administration**]: Security, integrity, and protection.

General Terms
Design, Security, Theory.

Keywords
Access Control, Collaboration, Trust Management, GTRBAC

1. INTRODUCTION
A global information infrastructure connects remote parties worldwide through the use of large scale networks, relying on application level protocols and service, such as recent Web service technology. Execution of activities in various domains, such as shopping, entertainment, business and scientific collaboration is increasingly based on the use of remote resources and services. The interacting parties in such application domains may be strangers or unknown to each other prior to getting connected. In a dynamic collaboration among remotely located parties coming from different security domains with no pre-existing relationship, the attributes of the parties serve as their identity. The attribute-based identification of collaborating parties is important to determine their authorization over each other's local resources. The party owning the resource or the service provider managing the service, specify the authorization of remote users based on their attributes and the requesting users are responsible for proving possession of the required attributes by providing relevant credentials. A user's credentials may include certificates issued by third parties or recommendations made by other users [5]. A user may possess multiple credentials certifying different attributes of the user. However, such credentials may not be accepted by the service provider with the same degree of trust. For example, a service provider may have a lesser degree of trust in the US citizenship attribute of a remote user if it is substantiated by the user's driving license as opposed to the passport. In addition, the degree of trust in a credential for verifying certain attributes of its possessor also depends on the trustworthiness of the party issuing the credential. In particular, in a distributed environment with no central certification authorities, all credential issuers may not be trusted to the same extent [3, 5, 4, 6] and consequently, the assertions made in the issued credentials may fail to certify the claimed attributes of the user with the desired degree of trust.

Ensuring secure and authorized access to remote services and information resources in a dynamic collaborative environment is a challenging task. Two key issues that need to be addressed in this regard are: specification of access control requirements and trust management. Specification of access control requirements for dynamic collaboration is a challenging problem mainly because of the limited or lack of knowledge about remote users' identities and affiliations. The access control policies and constraints defining users' authorization over remote resources and services need to be specified in terms of the attributes and properties of the users rather than their actual identities. Moreover, the criteria for validating the attributes of the remote users should also be specified as part of access control requirements. Various models for attribute based access control have been proposed in literature [14, 15]. However, such models do not distinguish users based on the trustworthiness of their credentials. Moreover, these models do not capture the context-dependent, in particular time-dependent, authorization constraints required in many service based applications [1, 6, 17].

DIM'05, November 11, 2005, Fairfax, Virginia, USA.

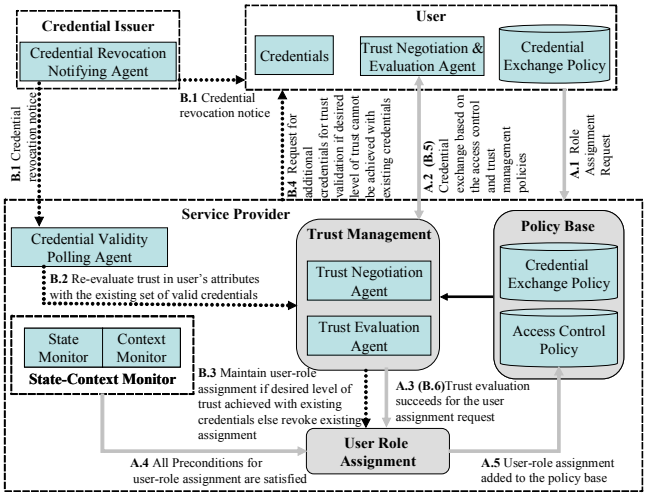

Figure 1. Proposed framework for access control and trust management.

Trust management, in the context of dynamic collaboration, not only involves validation of user's attributes for secure interoperation but also prevents the collaborating parties from making unauthorized inferences about each others sensitive attributes and policies. Trust-based validation of user's attributes is particularly important in a collaborative environment in which there is no central authority which every body trusts for credential certification [3, 4], rather assertions about user's attributes are made by the parties who may not be completely trusted by the service provider [5].

In this paper, we propose an agent-based framework for access control and trust management in a distributed collaborative environment. The framework uses Generalized Temporal Role Based Access Control (GTRBAC) [1] model for specification of authorization and security constraints in a distributed environment supporting dynamic collaboration among parties virtually unknown to each other. The key components of the proposed framework are shown in Figure 1. Detailed description of these components and the technical issues involved in their implementation is presented in the following sections.

2. PROPOSED FRAMEWORK

Figure 1 shows the proposed framework for access control and trust management in a distributed collaborative environment. In this framework, the service provider domain employs various agents for determining the authorization of external users based on their attributes. The attributes required for accessing a given resource are specified in the access control policy of the service provider. In the proposed framework, access to the requested resource/service cannot be granted unless the user proves possession of the required attributes with certain minimum degree of trust specified in the access control policy. The trust evaluation agent in the service provider domain is responsible for evaluating the degree of trust in the attributes claimed by the user based on the user's supplied credentials. In case the user's claimed attributes cannot be certified with the desired level of trust from the given credentials, the trust evaluation agent either rejects the user access request or asks for additional credentials from the user to increase the service provider's trust in the user attributes. Once the user's attributes are certified, the access request is forwarded to the user-role assignment module for further processing. In order to prevent disclosure of sensitive credentials and policies to untrusted parties during the attribute certification phase, a trust negotiation protocol is followed to ensure that credential exchange does not reveal any unauthorized information about the interacting parties. The trust negotiating agent in the service provider domain interacts with its counterpart in the user domain to perform such negotiations for trust establishment.

The policy base in the service provider domain stores the credential exchange and access control policies. Credential exchange policy drives the trust negotiation process with the external user. The access control policy is specified using extended GTRBAC model supporting attribute-based authorization specification. In addition, various context-related constraints are incorporated in the basic GTRBAC model [1] to support context-aware policy specification. A detailed discussion on the extended GTRBAC model is given in Section 3. In the GTRBAC abstraction, access to a service or information object by a user depends on user's authorization over the role to which such service or information object is assigned. For accessing the requested service or information object, the user must activate the corresponding role. A user can activate only those roles which are assigned to the user in the GTRBAC policy [1]. For assuming unassigned roles, a user first needs to request for role assignment. The request for role assignment is evaluated based on the degree of trust in user's attributes as discussed above and various environmental and contextual parameters specified in the access control policy. The user-role assignment module in the service provider domain is responsible for assigning external users to the requested role after ensuring that the user credentials have been certified with the desired degree of trust and all the access control constraints and preconditions for the requested role assignment are satisfied. After the role assignment, a credential is issued to the user certifying user's authorization over the requested role. The assignment of a user to a given role remains valid for a certain time interval specified in the access policy and need to be renewed after expiry of this time interval. In addition, expiration or revocation of some of user's credentials used in trust management phase may also lead to cancellation of the corresponding user-role assignment. The polling agent in the service provider's domain periodically checks the validity of users' credentials by polling the corresponding credential issuers. If a credential is found to be revoked or expired then the user is asked to provide alternate credentials for meeting the desired level of trust in user's attributes. In case, the user fails to provide such alternate credentials, the corresponding user-role assignment is cancelled.

Figure 1 also shows the sequence of messages exchanged between different entities and agents of the user, service provider and credential issuer domains for role-assignment. The messages exchanged for new role assignments are prefixed with the letter A and the messages for reevaluation of an existing role assignment due to credential expiration or revocation are prefixed with B. A detailed description of the contents of these messages is omitted because of space limitation.

Table 1. GTRBAC policy schema for trust-based access control management.

No.	Specification Type	Syntax
1	Role Definition	*RoleDef* → **Role** *r* {*Attribute*} [**EnableIntervals** I^{r}_{en}][**EnableDuration** D^{r}_{en}] [**ActiveIntervals** I^{r}_{act}] [**TotalActiveDuration** $D^{r}_{act\text{-}total}$] [**PerSessionDuration** $D^{r}_{act\text{-}session}$] {**UserAssignmentConstraint** C*ondition*} {**ValidationConstraint** *Condition*} {**ActivationConstraint** *Condition*}
2	Attribute Definition	*Attribute* → **Attribute** aname value *CredentialSet*
3	Credential-set	*CredentialSet* → *Credential* {*Credential*}
4	Condition Definition	*Condition* →*TrustEvalCondition* | CredentialExpirationCondition | *AssignSoDCondition*| *EnableSoDCondition* | *ActiveSoDCondition* | *TimeCondition* | D*ependenceCondition* | *CardinalityCondition* | *ContextCondition* | *Condition LogicOp Condition* | !*Condition* *TrustEvalCondition* → eval-trust(attrib-name, user-cred-set) ≥ $\text{threshold}_{\text{attrib-name}}$ *CredentialExpirationCondition* → ∀c∈ user-cred-set, exp-time(c) ≥ end-point(I^{ur}_{assign}) *AssignSoDCondition* → ∀*r'*∈ role-assign-sod(*r*), ∀*u*∈ USER, !ur-assigned(*u*,*r'*) *EnableSoDCondition* → ∀*r'*∈ role-enable-sod(*r*), !enabled(*r'*) *ActiveSoDCondition* → ∀*r'*∈ role-active-sod(*r*), ∀*u*∈ USER, !ur-activated(*u*,*r'*) *TimeCondition* → time Relation *Number* *DependenceCondition* → ur-assigned(*u'*,*r'*, Δ*t*) | enabled(*r'*, Δ*t*) | r-activated(*r'*, Δ*t*) | ur-activated(*u'*,*r'*, Δ*t*) *CardinalityCondition* → #u-assign(*r'*) *Relop Number* | #u-active(*r'*) *Relop Number* | #r-active(*u'*) *Relop Number* | #role-enabled *Relop Number* *ContextCondition* → ContextName.Value *Relop String* | ContextName.Value *Relop Number*
5	Role-Assignment request	*UserRoleAssignRequest* → **Request** Assign **User** *u* **Role** *r* [**AssignInterval** I^{ur}_{assign}] [**AssignDuration** D^{ur}_{assign}]{*CredentialSet*}
6	Role-Activation request	*UserRoleActiveRequest* →**Request** Activate **User** *u* **Role** *r* {**UserContextParam** *context*}
7	Trigger Definition	*TriggerDef* → {*Event*} ⇒ *Event* **after** Δ*t*
8	Event Types	*Event* → *enable r* |*disable r* |*assign r* to *u* |*de-assign r* to *u* | *activate r* for *u* | *de-activate r* for *u* | *change* in trust-level of *u*'s *attribute*(s)
9	Relations	*RelOp* → < | > | = | ≥ | ≤

3. ACESS CONTROL POLICY

The access control policy in the proposed framework is specified using the extended GTRBAC model. The extensions to the basic GTRBAC model include: i) addition of attribute specification for determining qualification of users for role assignment, and ii) inclusion of context based constraints in the policy specification for enabling context-aware access control in distributed environment. These extended constraints in conjunction with the basic features of GTRBAC can be used to model the domain specific access control policies for supporting dynamic collaborations in a distributed environment.

The basic GTRBAC model, which is a temporal extension of role based access control model [18], consists of following four components. a set of users, a set of roles, a set of permissions, and a set of sessions. A user is a human being or a process within a system. A role is a collection of permissions. A permission is an access mode that can be exercised on a particular object or resource. A session relates a user to possibly many roles and allows the user to access all permissions associated with such roles. A key aspect of the GTRBAC model is the notion of states of a role. In GTRBAC, a role can assume one of the three states: *disabled*, *enabled*, and *active*. A role is *enabled* if a user can acquire the permissions assigned to it. An *enabled* role becomes *active* when a user acquires the permissions assigned to the role in a session. By contrast, a *disabled* role cannot be activated by any user. Therefore, constraints on enabling of roles specify when roles can actually be activated by users. The GTRBAC model provides a complete framework for specification of temporal constraints on all events related to user-role and role-permission assignment, role enabling/disabling, and user-role activation.

Table 1 shows the specification schema of the GTRBAC policy for trust-based access control management. This specification schema is adapted from role-based policy specification for CSCW

systems proposed by Ahmed et. al. [2]. In the schema definition of Table 1, [ ] denotes optional constraints and { } represents zero or more occurrences of attribute terms or constraint expressions. The syntax of a role specification with associated attributes and constraints is shown in Row 1 of Table 1. In this definition, the term {*Attribute*} denotes the set of all attributes that must be possessed by a user for role assignment. The attribute definition in the specification schema of Table 1 includes attribute name, value, and a set of credentials. A user may provide a subset of these credentials to validate his/her claim of possessing the corresponding attribute. However, the degree of service provider's trust in this claim may vary depending on the type and number of credentials supplied by the user and the trustworthiness of the authority issuing these credentials. The function eval-trust listed in Table 2 is used to evaluate the degree of trust in a user's attribute with a given set of credentials.

In addition to attribute specification, various constraints on user-role assignment, role enabling, and activation are also included in role definition. These constraints are explained below.

3.1 Interval and Duration Constraints

GTRBAC allows specification of temporal constraints on different role-related events. The interval expression *I* in the role definition denotes a single or a set of intervals during which the corresponding role enabling or activation event can occur. In case the interval expression is not specified in the role definition, the corresponding event can occur any at any time. Duration constraints are used to specify duration for which enabling, assignment, or activation of a role is valid. When any of these events occur, the duration constraint associated with the event validates the event for the specified duration only. The variable D^r_{en} in the definition of role *r* specifies the maximum duration for which role *r* can be enabled within the time interval I^r_{en}. For role activation, duration constraints can be defined on a per-session basis as well as on the aggregate duration of all sessions in which the given role is active. The per-session duration limits the activation duration of role *r* in a single session to $D^r_{act\text{-}session}$. The aggregate duration constraint in the role definition implies that, within the activation interval I^r_{act} the total activation duration of the corresponding role in all sessions cannot exceed the maximum duration $D^r_{act\text{-}total}$. The duration constraint for user-role assignment is specified in the role assignment request as shown in Row 5 of Table 1. In case no duration constraint is specified in the role definition or assignment request, the corresponding event remains valid until it is disabled by some other means e.g., by a trigger.

3.2 User-Role Assignment Constraints

These constraints specify various security requirements related to assignment of a given role to users. These security requirements may include establishment of an acceptable degree of trust in user's claimed attributes, validation of all credentials supplied by the user, satisfaction of static (assignment time) separation of duty, cardinality, dependence, and other context-dependent constraints. The trust establishment condition, listed in Row 4 of Table 1, implies that a user's claim of possessing all relevant attributes necessary for role membership need to be proved with the acceptable level of trust. The parameter 'threshold$_{attrib}$' in the trust evaluation condition specify the acceptable level of trust for a particular attribute. The level of trust is evaluated for each attribute using the function eval-trust with the given set of user's credentials. For a user to qualify for role membership, the computed level of trust in each of the required attributes must be greater than or equal to the corresponding threshold value. In addition all the credentials used in evaluating trust in user's attributes must remain valid for the requested role assignment duration. The static (assignment time) separation of duty (SoD) constraint, listed as *AssignSoDCondition* in Row 4 of Table 1 prevents assigning two conflicting roles to the same user. The *dependence condition* in the role specification is used to specify the order in which the role assignment event must occur. For instance, a user must have a valid assignment for a certain role in order to qualify for the requested role assignment. These dependency constraints are specified using the role-assignment, role enabling, and role activation predicates which are listed in Table 2. The cardinality condition in the role assignment specification defines an upper bound on the number of users to which a given role can be assigned. The schema definition of Table 1 supports various form cardinality constraints including maximum number of assignments for a single role and maximum number of roles that can be assigned to a single user. Additionally, preconditions for role assignment based on user and/or environmental context can be specified in the role definition. Typical context parameters include time, location, system load etc.

3.3 Role Validation and Activation Constraints

Role validation constraints must be satisfied throughout a user's membership in a given role [2]. If these constraints are not satisfied, the user's assignment to the role is cancelled. In the access control management framework discussed in this paper, role validation constraints may be violated because of the revocation of user credentials or change of user context parameters. The role validation constraints are specified in the GTRBAC policy using event trigger expressions as shown in Row 7 of Table 1.

Role activation constraints specify the pre-conditions for activation of roles by the authorized users. These pre-conditions are checked at the time of role-activation request and must hold throughout the activation duration of the requested role. The activation constraints for a given role may include dynamic SoD constraints, dependence constraints, and cardinality constraints. The dynamic SoD constraint, listed in Row 4 of table 1, prohibits concurrent activations of two or more roles by the same user. Similarly, other types of SoD constraints can be specified for role activation using the role activation predicates listed in Table 2. The dependence constraints are used to specify the order in which roles need to be activated. Cardinality constraints limit the total number of concurrent activations of a given role by the same user or by multiple users. In addition activation of a role can be constrained based on the user or environment context.

Table 2. Functions and predicates used in access control policy specification schema.

Function/Predicate	Semantics
eval-trust(attr, C^u)	Returns the trust value for user attribute 'attr' with the given credential set C^u.
exp-time(c)	Returns the expiration time of the credential 'c'.
end-point(I)	Returns the end-time of an interval I.
role-assign-sod(*r*)	Returns the role-set R_{assign}= { *r*'| *r'* and *r* cannot be assigned to the same user simultaneously}.
role-enable-sod(*r*)	Returns the role-set R_{enable} = { *r*'| *r'* and *r* cannot be enabled concurrently}.
role-active-sod(*r*)	Returns the role-set R_{active}= { *r*'| *r'* and *r* cannot be activated by same user concurrently }.
u-assign(r)	Returns the set of users assigned to role *r*.
u-active(r)	Returns the set of users assuming role *r* in their ongoing sessions.
r-active(u)	Returns the set of roles being activated by user *u*
ur-assigned(*u*,*r*)	Returns true if role *r* is assigned to user *u*.
ur-assigned(*u*,*r*,Δ*t*)	Returns true if role *r* is assigned to user *u* for at least Δ*t* time units.
Enabled(*r*)	Returns true if role *r* is in enabled state.
Enabled(*r*, Δ*t*)	Returns true if role *r* is in enabled state for at at least Δ*t* time units.
ur-activated(*u*,*r*)	Returns true if role *r* is active in *u*'s ongoing session
ur-activated(*u*,*r*, Δ*t*)	Returns true if role *r* is active in *u*'s ongoing session for at at least Δ*t* time units.
ur-activated(,*r*, Δ*t*)	Returns true if role *r* is active in any user's session for at at least Δ*t* time units.

<Role PC Member **EnableInterval** [01\06\05, 06\31\05] **PerSessionDuration** 30 mins
<**Attribute** *Type*: Academic Qualification *Value*:Ph.D
<**Credential** Academic Degree **\Credential**> **\Attribute**>
<**Attribute** *Type*: Professional Qualification *Value:* Faculty or
Academic Researcher or Industry Researcher
<**Credential** Employer Certificate **\Credential**> **\Attribute**>
<**Attribute** *Type:* Research Standing Value: good
<**Credential** Publication Index **\Credential**>
<**Credential** Citation Index **\Credential**>
<**Credential** Patent Index **\Credential**> **\Attribute**>
<**Attribute** *Type:* Research Area Value: Database Systems
<**Credential** Publication List **\Credential**> **\Attribute**>
<**Attribute** Type: Membership Status Value: Regular or
Senior, or Fellow <**Credential** Member Certificate **\Credential**>
\Attribute>
<**UserAssignmentConstraints**
eval-trust(Academic Qualification, {Academic degree}) +
eval-trust(Professional Qualification, {Employer Certificate}) ≥ 0.9
eval-trust(Research Standing,{Publication index, Citation index, Patent Index})≥0.6
#project(publication-list, Database) ≥ 10
exp-time(Member Certificate) ≥ 06\31\05
!ur-assigned(*this*.user, author)
#u-assign(PC member) ≤ 30 \ **UserAssignmentConstraint s>**
<Validation Constraints
expired(*this*.user,Membership)→de-assign(this.user, PC Member)
ValidationCostraints>
<Activation Constraints
ur-activated (*this*.user, PC Member) → activate Reviewer for *this*.user
#u-active(PC member) ≤ 10 **Activation Constraints**>
Role>

Figure 2. GTRBAC-based specification of PC Member role.

3.4 Example

Figure 2 shows an example of the access control policy specification of a Program Committee (PC) member role of a conference. This role specification is based on the GTRBAC-based policy schema of Table 2. The PC member role is enabled during the interval [01\06\05, 06\31\05] and can only be activated by authorized users during this interval. The role can be activated any number of times for at most thirty minutes. The attributes required for the membership of this role include: i) doctorate level academic qualification, ii) a university faculty or a researcher position in an academic institute or commercial organization, iii) 'good' research standing in Database Systems area, and iv) valid membership of the conference sponsoring organization. The credentials required for validation of these attributes are listed in the corresponding attribute definition. The academic and professional qualification attributes of the candidate user can be validated by the academic degree and the employer certificate of the user. However, the aggregate trust in the validity of these attributes must be greater than 0.9. This trust condition, specified as one of the user assignment constraints in Figure 2, indirectly determines the academic and professional standing of the candidate user based on the user's academic and professional profile. The research standing of the candidate user can be evaluated from the user's publication index, citation index, or patent index. For a user to qualify for good research standing in the database area the trust value computed from these credentials must be greater than 0.6. In addition, the user should have at least 10 publications in the database area. This condition is specified in the definition of the PC member role using the composite function #project with the user's publication-list and the string 'Database' as arguments. With these arguments, the function project returns all the database related publications from the given publication list and the '#' operator returns the count of the projected list. A PC member cannot be the author of any paper submitted to the same conference. This is an assignment time SoD constraint specified in the role definition using the negated predicate !ur-assigned(*this*.user, author). Finally, the total number of user assignments for the PC member role must not exceed thirty. This is specified in the role definition using the cardinality constraint #u-assign(PC member) ≤ 30.

The trigger-based validation constraint, *expired* (*this*.user,Membership) → *de-assign*(this.user, PC Member), specified in the definition of PC Member role, implies that expiration of a user's membership from the conference sponsoring organization will result in the cancellation of the user's assignment to the PC member role. There are two activation constraints defined for the PC member role. The first constraint defines the activation time dependency between the PC member and reviewer roles. This dependency implies that activation of the PC member role by a user must be followed by the activation of the Reviewer role by the same user. The second activation constraint specifies the activation cardinality of the PC member role, implying that at most ten users can activate the PC Member role at any given time.

3.5 XML Specification

The features of the extended GTRBAC model discussed above can be specified using XML, which has become a default standard for sharing and dissemination of information contents and policies over the Internet. We have developed an XML-based framework, called X-GTRBAC [19], for implementing the semantics of extended GTRBAC model. This framework allows specification and enforcement of access management policies supporting attribute-based authorizations and context-aware access control requirements in a dynamic collaborative environment [19]. X-GTRBAC allows specification of all the elements of the GTRBAC model. These specifications are captured through a context-free grammar called X-Grammar. X-Grammar allows the composition of XML-based policy documents using a vocabulary of various policy sheets and definitions, which are used to define users, roles, permissions, and user-to-role and permission-to-role assignments in the GTRBAC policy. The grammar also captures the temporal constraint expressions of the GTRBAC model, such as constraints on role enabling, activation and assignment, and non-temporal contextual constraints of the extended GTRBAC model. The detailed specifications of X-GTRBAC framework can be found in [19].

To incorporate the attribute-based authorization in the X-GTRBAC framework, we have integrated the support for the Security Assertion Markup Language (SAML) standard [20] into X-GTRBAC specification. SAML provides a message exchange protocol for communicating attributes and credentials among different autonomous parties. However, the protocol needs to be tied to authentication and authorization mechanisms to support secure information accessibility in the distributed collaborative environment. The X-GTRBAC framework provides such mechanisms. Detailed discussion on the transformation of SAML assertions into X-GTRBAC specification can be found in [21].

4. TRUST MANAGEMENT

In this paper, we consider two key aspects of trust management, including: i) trust evaluation, and ii) trust negotiation. The former deals with assessment of trust in a user's claimed attributes for determining the user's authorization over the requested resource. The latter involves establishing trust between the collaborating parties (user and service provider) for disclosure of sensitive credentials and policies.

4.1 Trust Evaluation

As discussed above, the attributes required for accessing a given resource are specified in the access control policy of the resource owner and the requesting users are responsible for proving possession of the required attributes by providing relevant credentials. The trust evaluation agent in the proposed framework of Figure 1 is responsible for verifying the claimed attributes of the user with the required degree of trust specified in the access control policy. The degree of trust by which a certain attribute of the user can be verified depends on the type of credentials provided by the user and the trustworthiness of the parties issuing such credentials. In the following we describe the key issues related to trust evaluation.

In a distributed collaborative environment with no central certification authority, the issuers of the credentials may not be trusted to the same extent [3, 4, 5, 7]. For instance, in a PGP-based *Web Of Trust* model individual users recommend other users to the service providers by signing the PGP keys of the recommended users. These recommendations may serve as credentials asserting recommenders' trust in certain attributes or

properties of the recommended users. However, the trustworthiness of the recommenders may vary depending on the relationship of the recommenders with the service provider [4, 5]. In the PGP model, the recommenders are classified into four distinct levels based on their trust-worthiness. These trust levels are listed below:

Full: the recommender is fully trusted to recommend another user.

Marginal: the recommender can be trusted to recommend/introduce another user, but, it is not certain whether the recommender is fully competent to make the recommendation.

Untrustworthy: the recommender should not be trusted to recommend another user, therefore any recommendation by such recommender should be ignored.

Don't know: There are no expressions of trust made about this recommender.

To compensate for the ambiguity of the above trust levels for recommendations, a service provider may specify the required number of recommendations by fully trusted and marginally trusted recommenders for verifying a given attribute of the recommended user [5]. In addition, recommenders may also have a varying degree of trust about the attributes of the recommended user [8, 6]. For instance, one recommender may *completely* trust the "good academic standing" of the recommended user. Another recommender may also assert the "good academic standing" of the same user but with *nominal* degree of trust. Therefore, the trust value derived from these recommendations may be different than the trust value embedded in the recommendations. A weighted average method can be used to evaluate the trust value for a given attribute of the user from multiple recommendations, where the weights may correspond to the numerical value of the trust assigned to the corresponding recommenders by the service provider.

Some application domains allow transitivity of trust in recommendations [4, 3, 6, 8]. For example, a service provider A trusts B as a recommender, and B trusts C as a recommender. C can forward his/her recommendation about a user D to B. Since C is trusted by B as a recommender, B accepts any recommendation made by C and can forward it to any service provider who trusts B as a recommender. However, D's level of trust evaluated by the service provider A may be different than the level of trust evaluated by the recommender B. Similarly, B and C may have a different degree of trust in D's attributes. Various models and protocols have been proposed for propagating trust in a distributed Web-based environment [6, 8, 4]. All of these models first explore the network of recommenders and users to find possible paths from the service provider to the end user. Then an aggregate function is used to combine the trust values computed from each path into a single value. The main difference between these trust propagation models is the use of different trust metrics for classification of recommendation agents and the aggregate functions for evaluating the final trust value.

An important issue not adequately addressed in current literature is of trust reevaluation when one or more of the user's credentials are revoked. Revocation of a user's credentials may not necessarily imply that the user is malicious and cannot be trusted. A credential issuer may revoke the credentials of a given user for a number of reasons, including expiration of the validity time of the user, change in the environmental or user context. For instance, a user may move out of state and therefore the residence attribute of the user in his/her driving license does not remain valid. Similarly, the affiliation of the user with the credential issuer may also change. For instance, if an employee leaves the company then the credentials issued by the old company become invalid and are revoked. The revoked credentials might have been used by the user to prove possession of certain attributes for which the credentials were not issued primarily. For instance, a driving license primarily certifies its owner's competence for driving; however, it can be used by a user to verify his/her US residence status. Therefore, revocation of the driving license for reasons other than change in the residence status of a user should not lead to cancellation of the user's membership to a role which can only be assigned to US residents. In this case the user should be given a chance to prove his/her residence status using alternate credentials such as passport, tax returns, or utility bills. A major problem in this regard is the selection of alternate credentials and the re-establishment of the trust between the collaborating parties in a timely manner. This problem can become more challenging in the distributed *Web Of Trust* environment in which evaluation of the trust level of alternate credentials may require discovering a new chain of recommendation agents. In this environment, the validity of a recommendation for a user may also get affected if any of the recommender in the transitive chain of recommendations leaves the *Web Of Trust*. Although in this case the recommendation for the user is not revoked by any recommender, the breaking of the recommendation link between the service provider and the end-user requires trust re-establishment.

4.2 Trust Negotiation

As discussed above, credential exchange facilitates in establishing mutual trust between the service provider and the end user that do not have any pre-existing relationship. The disclosure of credentials to the requested party is governed by a credential exchange policy. The negotiating parties rely on the disclosure of credential exchange policy to learn each other's access control requirements. However, the credential exchange policy may itself contain sensitive information and disclosing its contents unconditionally may leak valuable business information which may be used against the interests of either one or both of the negotiating parties [9, 11].

Various automated trust negotiation strategies have been proposed to prevent unauthorized disclosure of credential exchange policy or leakage of any information that may be used by the negotiating party for inferring about the possession of sensitive credentials by the opposite party [9, 10, 12, 13, 11, 12]. These strategies rely on iterative disclosure of credentials and associated policies to ensure safe negotiation. In such negotiations, credentials are unconditionally accepted, i.e., if a requested credential is disclosed then the requesting party accepts this credential with the highest level of trust. However, in a decentralized collaborative environment such as *Web Of Trust*, a credential may not be completely trusted. Therefore, disclosure of a requested credential may not necessarily satisfy the access requirements for the target credential or the requested resource of the other party. To ensure safety in automated trust negotiation in a decentralized collaborative environment, the existing trust negotiation strategies need to be adapted to allow the negotiating

party to evaluate the trust level of the credentials disclosed by the opposite party before revealing any information about undisclosed credentials. For such adaptation, first the negotiation policies need to be tailored for specification of the credentials along with the acceptable trust values, required for continuation of trust negotiations. The GTRBAC based formalism discussed in Section 3 can be used for specification of such negotiation policies. In addition, the existing negotiation strategies need to be revised for disclosure of new credentials and policies based on the trust level of the credentials supplied by the negotiating parties.

As discussed in Section 4.1, the level of trust assigned to a given credential also depends on the trustworthiness of its issuer. During the trust negotiation, failure of a credential to meet the acceptable trust level may enable the party supplying the credential to infer information about the trustworthiness of various credential issuers and collaborators. This may be confidential and may affect the business relationship among the collaborating parties. The problem of inference in trust negotiation has been studied in literature in the context of possession-sensitive and attribute-sensitive credentials [12, 10, 11]. Yu and Winslett [10] have proposed a policy migration technique for preventing inference about possession-sensitive credentials during trust negotiation. Winsborough and Li in [11] have also proposed a strategy based on credential combination-hiding for preventing inference about sensitive credentials. These techniques need to be analyzed in a decentralized collaborative environment for prevention of inference about the trustworthiness of collaborating parties during trust negotiation.

5. Conclusion

In this paper, we have proposed a framework for access control management in a distributed environment supporting dynamic collaboration between remote users and service providers. In this framework, the authorizations of remote users are determined based on their attributes. The attributes required for accessing a given resource are specified in the access control policy of the service provider and the users are responsible for proving the possession of the required attributes for the requested resource by providing relevant credentials. We have discussed several issues related to validation of the user supplied credentials for ensuring secure and authorized information access. In particular, we have discussed trust-based validation of credentials in a decentralized environment with no central certification authorities which everybody trusts. Another important issue, discussed in the context of establishing secure collaboration between remotely-located parties, is preventing inference about collaborating parties policies and sensitive credentials.

6. ACKNOWLEDGEMENTS

The work reported in this paper has been partially sponsored by the National Science Foundation under the ITR Grant No. 0428554 "The Design and Use of Digital Identities" and by the sponsors of Center for Education and Research in Information Assurance and Security (CERIAS) at Purdue University.

7. REFERENCES

[1] Joshi, J. B. D., Bertino, E., Latif, U., and Ghafoor, A. Generalized temporal role based access control model. *IEEE Transactions on Knowledge and Data Engineering*, 17, 1 (Jan. 2005), 4-23.

[2] Ahmed, T. *Policy Based Design of Secure Distributed Collaboration Systems*. Ph. D. Thesis, University of Minnesota.

[3] Grandison, T., and Sloman, M. A survey of trust in Internet applications. *IEEE Communications Surveys*, Fourth Quarter, 2000, 2-14.

[4] Rahman, A.-A., and Hailes, S., A distributed trust model. In *Proceedings of the Workshop on New Security Paradigms*, 1997.

[5] Rahman, A.-A. The PGP trust model. *The Journal of Electronic Commerce*, 1997.

[6] Ziegler, C. N., and Lausen, G. Spreading activation models for trust propagation. In *Proceedings of the IEEE International Conference on e-Technology, e-Commerce, and e-Service*, 2004.

[7] Mass, Y., and Shehory, O. Distributed trust in open multi-agent systems. Trust in Cyber Societies, *LNAI, R. Falcone, M. Singh, and Y.-H Tan Edition*, 2001, 159-173.

[8] Richardson, M., Agrawal, R., and Domingos, P. Trust management for the semantic web. In *Proceedings of the International Semantic Web Conference*, 2003.

[9] Yu, T., and Winslett, M., A unified scheme for resource protection in automated trust negotiation, In *Proceedings of the IEEE Symposium on Security and Privacy*, May 2003.

[10] Yu, T., and Winslett, M., Policy migration for sensitive credentials in trust negotiation, *In Proceedings of the ACM Workshop on Privacy in the Electronic Society*, 2003, 9-20.

[11] Winsborough, W., Li, N. Safety in automated trust negotiation, In *Proceedings of the IEEE Symposium on Security and Privacy*, 2004, 147-160.

[12] Seamons, K.E., Winslett, M., and Yu, T. Limiting the disclosure of access control policies during automated trust negotiation. In *Proceedings of the Workshop on Privacy Enhancing Technologies*, 2002.

[13] Winsborough, W., and Li, N. Towards practical automated trust negotiation. In *Proceedings of the IEEE Workshop on Policies for Distributed Systems and Networks*, 2002, 92-103.

[14] Li, N., Mitchell, J. C., and Winsborough, W. Design of a role-based trust-management framework. In *Proceedings of the IEEE Symposium on Security and Privacy*, 2002, 114-130.

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

[16] Bertino, E., Ferrari, E., and Atluri, V. The specification and enforcement of authorization constraints in workflow management systems. *ACM Transactions on Information and System Security*, 2, 1, (1999), 65-104.

[17] Yu, J. Dynamic web service invocation based on UDDI. In *Proceedings of the IEEE International Conference on E-*

Commerce Technology for Dynamic E-Business, 2004, 154 – 157.

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

[19] Bhatti, R., Ghafoor, A., Bertino, E., and Joshi, J. B. D. X-GTRBAC: an XML-based policy specification framework and architecture for enterprise-wide access control. *ACM Transactions on Information and System Security*, 8, 2 (May 2005), 187-227.

[20] OASIS XML-Based Security Services Technical Committee (SSTC). Security Assertion Markup Language (SAML). Technical Report, http://xml.coverpages.org/saml.html.

[21] Bhatti, R., Bertino, E., and Ghafoor, A. An integrated approach to federated identity and privilege management in open systems. Accepted for publication in the *Communications of the ACM*. Also available as CERIAS Technical Report TR 2005-42, https://www.cerias.purdue.edu/tools_and_resources/bibtex_archive/archive/2005-42.pdf.