CERIAS Tech Report 2002-60
Authorization Based on Evidence and Trust
by B Bhargava, Y Zhong
Center for Education and Research
Information Assurance and Security
Purdue University, West Lafayette, IN 47907-2086

Authorization based on evidence and trust*

Yuhui Zhong Bharat Bhargava
Center for Education and Research in Information Assurance and Security
and
Department of Computer Science
Purdue University
West Lafayette, IN, U.S.A.

Abstract

Developing authorization mechanisms for secure information access by a large community of users in an open environment is challenging. Current research efforts grant privilege to a user based on his/her properties that are demonstrated by digital credentials (evidences). Holding credentials does not necessarily certify that the user is trustworthy. We use trust to characterize the possibility that a user will not carry out harmful actions. Authorization based on trust as well as evidence makes access control adaptable to users' behaviors. The research requires: a suitable authorization mechanism that can incorporate the evidence and the trust, appropriate representations of evidence and trust so that their manipulation can be automated. In this paper, we present a trust-enhanced role-mapping server, which can cooperate with RBAC (Role-Base Access Control) mechanisms for authorization based on evidence and trust. The effort of formalizing trust and evidence is discussed.

1. Introduction

Two main classes of security services are needed to build a secure Internet infrastructure: access control services and communication security services [11]. The access control services ensure that information is accessed or manipulated only by authorized persons. Research is needed to develop authorization mechanisms for a large and open community of users. In such an environment, prior knowledge about a user may not exist [16]. For authorization, the permission set for each user must be determined. Evidence (or credential [13]) and trust are two main ideas that can be used to accomplish this. Current research efforts grant privilege to a user based on her properties that are demonstrated by digital credentials (evidences) issued by third parties [1], [14]. Holding credentials does not certify that the user is trustworthy. The impact of users' behaviors on their trust with system needs to be quantified. Furthermore, the reliability of evidence/credentials from different issuers might be different. For example, evidence that is provided by issuer A is fully trusted while evidence that is provided by issuer B is partially trusted. The term *trust* is used to characterize the probability that a user/issuer will not carry out harmful actions [22]. Authorization based on evidence as well as trust makes access control adaptable to users/issuers' behaviors. The research requires: An appropriate representations of evidence and trust so that their manipulation can be automated, a suitable authorization architecture that can incorporate the evidence and the trust, integration of this scheme with existing access control mechanisms. We investigate these issues and propose a trust-enhanced role-mapping server architecture, which can cooperate with RBAC (Role-Base Access Control) mechanisms for authorization based on evidence and trust. We introduce evidence and trust briefly.

Evidence: Evidence (also called credential) is a statement about certain properties of an entity (called subject) issued by the issuer. Evidence can come from internal or external sources. Evidence can be information stored in local database (user name and password) or public key certificate (e.g. X.509 V3) [10][4], digitally signed document (e.g. PICS rating) [17], etc. Evidence with different forms or from different issuers should be trusted differently. Our research effort investigates the following issues: How to associate different trust degrees with evidences? What factors affect the trust of evidence and how to determine the trust degree of certain evidence? How to accommodate different formats of evidences in one framework?

Trust: Trust plays an important role when a user makes decisions with uncertainty and incomplete information in applications such as e-commerce and virtual communities [18]. Trust

*This research is supported by CERIAS and NSF grants CCR-9901712 and CCR-0001788.

is a subjective degree of belief [6]. The aspects forming the trust and the weights of the aspects might be different, for different entities or one entity in different environments. Therefore, different observers may have different perceptions of the same entity's trustworthiness. Trust is formed mainly by two ways: (1) get opinions from third parties (i.e. second hand information). Because trust is not transitive [3], the opinions from third parties cannot be directly used. (2) summarize prior interactions (i.e. first hand information). The research issues include: How to represent trust, which is subjective and multi-faceted, in a computational model? How to make trust assessment based on both first-hand and second-hand information?

The rest of this paper is organized as follows. Section 2 introduces related research. Section 3 presents the fundamental concepts in our system and their formal definitions. The architecture of *role server* is described in section 4. The algorithms and implementation are in section 5. We focus on the role-assignment policy language and the algorithms evaluating the reliability of evidence and role-assignment policies. The paper is concluded in section 6.

2. Related work

Authorization in an open environment: It is a challenging authorization problem to controlling access for users in an open environment, because the user is not necessarily known by the system when he/she makes the access request. Several research efforts have been undertaken in this area. One direction is *trust management* [13][14][22]**.** A trust management system provides a language allowing system administrators to define authorization policies based on credentials, and an engine to enforce the authorization polices. These systems design their own access control mechanisms instead of taking advantage of existing ones.

Another direction of research divides the authorization problem into two sub-problems: (1) determine the permission set of a user (2) enforce access control by using existing mechanisms like RBAC [1][16]. These Approaches have the advantage of easy integration with existing systems. Our research effort is in this direction. Users' permission sets are determined based on evidence and trust, which distinguished our work from others that determine users' permission set only according to evidence/credential. Furthermore, reliability of evidence is considered in our system.

Trust Models: Several researchers have proposed algorithms to summarize opinions from third parties' trust opinions. The summarization includes evaluating an opinion from an entity, or combining opinions from different entities [2][3][20]. Few research efforts have been done to quantify trust based on direct-experience. Because personal experience plays an important role when forming trust opinion in real life, we consider first-hand information as well.

RBAC: RBAC has emerged as a promising technology for efficiently managing and enforcing security in large organizations [8][12]. A role is an entity with some semantics regarding the authority and responsibility. The authorization process is divided into two parts: role-permission mapping and user-role mapping. Role-permission mapping associates roles with permission set. User-role mapping assigns roles to users.

3. Concepts and formal definitions

The important concepts and their representations in our system are presented in this section,

3.1 Concepts

Evidence and credential: Evidence and credential are statements about some properties of a subject. We consider statements gained from outside of our framework as "credentials" and statements within our framework as "evidence".

Issuer's opinion about evidence: The current credentials do not provide a way for issuers to express their opinions towards the statements they make. When an issuer makes a statement, she is assumed to be 100% sure about it. This is not necessarily true in many cases. Issuer's opinion

about evidence characterizes the degree to which an issuer is sure about the statement he/she makes.

Reliability of evidence: The reliability of evidence represents the degree of truth of evidence from the point of view of the entity relying on the evidence. Reliability of evidence is a subjective concept. Different relying-party may have different perceptions of the same evidence. The reliability of evidence depends on issuer's opinion and relying-party's opinion towards the issuer.

Trust associated with an issuer and with a normal user: Trust associated with an issuer should be distinguished from that associated with a normal user. The former characterizes the trust to the evidence provided by the entity. The latter represents the trust of that fact that the entity's own behavior is co-operative.

Direct-Experience and recommendation: The opinions toward an entity from other entities are called "recommendations", which are second-hand information. The prior interactions between the observer and the percept are called "direct-experience". Direct-experience is first-hand information.

Trust environment: Trust is environment specific [6]. Different aspects of trust might be emphasized in different environments. The measurement of the same aspect of trust may vary in different environments. For example, if Alice is a doctor, Bob may trust her judgment on health problems. However, he may hesitate to buy the car recommended by her. How to represent environment and propagate trust in different environments is the issues we investigate.

3.2 Definitions and representations

Definition: An ***evidence type*** is a 2-tuple (*et_id, attrs*) where *et_id* is the identifier of this evidence type and *attrs* is a set of attributes. Each attribute is represented as a triple (attr_name, attr_domain, attr_type). Attr_type ∈ {opt, mand} which specifies whether the attribute can have a null value (attr_type="opt") or not (attr_type="mand"). Evidence type specifies information that is required by kind of evidences.

Example: (student, [{name, string, mand}, {university, string, mand}, {department, string, opt}]) is an evidence type. It indicates that "name" and "university" are required for this kind of evidence while "department" is optional.

Evidence type hierarchy: The whole set of evidence types forms an evidence type hierarchy as shown in figure 1. The first level of the hierarchy represents the two subsets of evidence types that we consider: *credential_evidence* and *trust_evidence*. Credential_evidence includes the set of all possible credential types recognized by the role server. Trust_evidence includes the set of all possible representations used by the role server to describe trustworthiness. Level 2 consists of access_credential, access_trust, testify_credential, and testify_trust. Access_credential and access_trust represent credential/trust related to normal user. Testify_credential and testify_trust

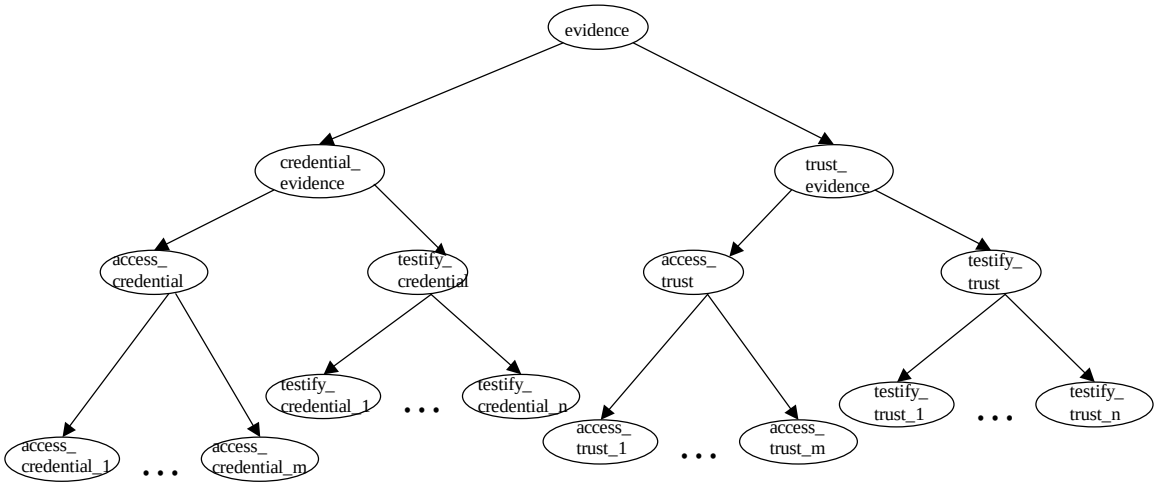

Figure 1 Evidence Hierarchy

is used for represent credential/trust related to issuer. The rest evidence types inherit one of them.

Definition: An ***evidence*** is a triple (*e_id*, *et_id*, *state*), where *e_id* is the identifier of this evidence, *et_id* is an evidence type identifier, *state* = (a1:v1, …, an:vn), where a1, a2, …, an are the names of attributes, v1, …, vn are their values. Evidence is an instance of an evidence type.
Example: (proof_of_Michael_as_a_student, student, {name: "Michael", university: "Purdue"}) is an evidence. The type of this evidence is "student". It proves that the holder of the evidence has certain specified properties that are required for this type of evidence: his name is "Michael" and his university is "Purdue".
The definitions of evidence type and evidence are similar to the credential models in [7].

Definition: ***Opinion*** is a triple (*b*, *d*, *u*) where b, d and u designate belief, disbelief and uncertainty respectively. They satisfy the equation: $b + d + u = 1$ $b, d, u \in [0, 1]$

Definition: Let w=(*b*, *d*, *u*) be an opinion. The ***probability expectation*** of w, denoted by E(w), characterizes the degree of truth represented by an opinion. E(w) is defined as: $E(w) = b + 0.5*u$
We assume that uncertainty about {belief, disbelief} can be split equally between the two states based on the principle of insufficient reason.

Definition: An ***evidence statement*** is a quadruple <*issuer, subject, evidence, opinion*>. *Issuer* is the entity, which provides the evidence. *Subject* is the entity to which the evidence refers. *Evidence* contains properties of the subject, which can be either credential or trust information in the form of evidence. *Opinion* characterizes the **issuer**'s belief towards the *evidence*.
Evidence statement is the most important abstraction in the role-mapping server architecture. The exchange of information among components is accomplished by using the evidence statement. Evidence statement provides a uniform view of different kinds of credentials and trust information. It associates credentials with different trust degree. Evidence statement makes it easy to adopt new type of credentials.

Role classification: To simplify the design and implementation, without loss generality, we classify roles into two categories and assume that there is no overlapping between these two categories.

- ***Access role category*:** A role belongs to *access role category* if its permission set includes particular types of access to one or more objects of the system. We call this kind of roles access roles. A normal user should hold certain access roles.
- ***Testifying role category*:** A role belongs to *testifying role category* if its permission set includes providing evidence for other entities. We call this kind of roles testifying roles. An issuer should hold certain testifying roles. The system only accepts the evidence from entities holding appropriate testifying roles specified in mapping policies. A testifying role has no permission to access the resources on the website (e.g. read or write documents). If a user needs both types of privileges, he/she has to get both access roles and testifying roles.

Representation of trust information: Evidence Statement is used to convey trust information. We distinguish trust as normal users (i.e. trust as access roles) from trust as issuers (i.e. trust as testifying roles) in section 3.1.

- **Trust as access roles:** Trust as access roles is represented as <*I, u, access_trust, opinion*>.*I* is a particular instance of issuer, which denotes the role-mapping server itself.*U* refers to the user. *Opinion* denotes how much role-mapping server believes the above statement.
  Access_trust is an evidence type whose semantic is that the user will co-operate and not defect. *Access_trust* contains three attributes <*s, c, i*>. The domains of the three attributes

are [0, 1]. Each attribute characterizes one aspect of harmful actions. The higher the value is the lower the probability is that a user will carry out such kind of harmful actions.

(1) Attribute s denotes the attempt to get unauthorized access.

(2) Attribute c characterizes the action of consuming enormous amount of resources.

(3) Attribute i represents the result of information leak or gather wrong information.

- **Trust as testifying roles:** Trustworthiness of a user as testifying role is represented as <*I*, *u*, *testify_trust*, *opinion*>. *I* is a particular instance of issuer, which denotes the role server itself. *U* refers to the user. *Opinion* denotes how much role-mapping server believes the above statement. *Testify_trust* is an evidence type whose semantic is that the user will provide accurate information of other users. *Testify_trust* contains one attribute <*t*> whose domain is [0, 1]. The higher the value, the higher an evidence provided by the corresponding user is trustworthy.

4. Architecture of trust-enhanced role-mapping server

We propose a trust-enhanced role-mapping server that can collaborate with a RBAC–enhanced web server for authorization in open environments as shown in figure 2. There are client, RBAC enhanced web server and trust-enhanced role-mapping server. The task of the role server is to

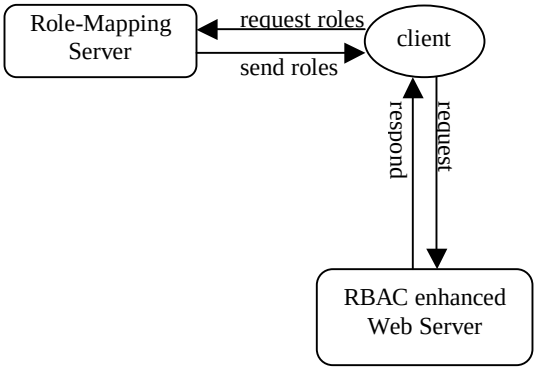

Figure 2 Authorization with Role-mapping Server

map users to roles based on evidence and trust. Clients obtain the roles from role-mapping server and present them to RBAC enhanced web server. Upon receiving a request from a client, the RBAC enhanced Web server checks if the user holds appropriate roles and sends back the object if the answer is true. The focus of this paper is in the role-mapping server, which maps users to roles.

In order to map a user to roles, the role-mapping server first collects credentials and transforms them to evidence statements, then evaluates the reliability of evidence based on evidence statement and issuer's trustworthiness, finally maps user to roles based on assignment policies, evidence/reliability users' trustworthiness.

A trust-enhanced role-mapping server consists of four components as shown in Figure 3. These components are based on the concepts discussed in the previous section. They exchange information using an evidence statement that we present in section 3.

The components are as follows:

Credential Management Component transforms different formats of credentials to evidence statements.

Evidence Evaluation Component evaluates the reliability of evidence statements.

Role Assignment Component maps roles to users based on the evidence statements and role assignment policies.

Trust Information Management Component evaluates user/issuer's trust information based on direct-experience and recommendations.

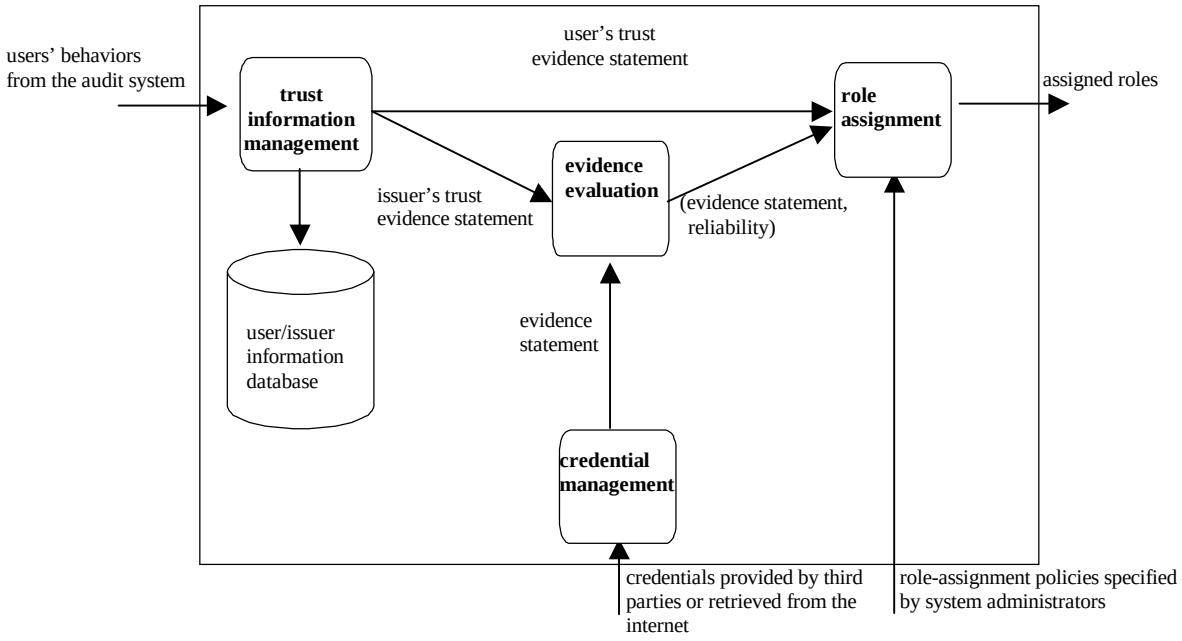

Figure 3 Trust-enhanced Role-Mapping Server Architecture

5. Algorithms and implementation

We have designed a role assignment policy declaration language to specify the requirements for assigning a role to a user. The algorithms to evaluate the reliability of evidence and role-assignment policies have been developed. We have implemented a prototype including **Evidence Evaluation** and **Role Assignment** and part of **Trust Information Management**. The prototype can be used to define evidence types, add users and roles, define role assignment polices, load evidences stored in a file and automatically assign roles to users based on role assignment polices and evidences. In this section, we present the algorithms in evidence evaluation and role assignment, and the policy declaration language. The research issues in trust information management and credential management are briefly discussed.

5.1 Evidence Evaluation

Evidence evaluation component determines the reliability of evidences from the role server's point of view. Reliability parameter indicates the degree to which the system believes that the corresponding evidence is true. It is a number $\in$ [0, 1]. The higher value means that the system believes that the associated evidence is more convincing. The reliability is computed on the basis of the opinion(s) included in the evidence statement(s) and the issuer's testify_trust.

We have designed an algorithm that uses the discounting operators proposed in [9] to evaluate the reliability of evidence. The ratio of belief to disbelief may affect the distribution of uncertainty. We plan to investigate this issuer in future research

Algorithm to evaluate reliability of evidence

Input: an evidence statement E_1 <*issuer, subject, evidence, opinion*$_1$>

Output: The reliability of the evidence statement RE (E_1)

Step1: Extract *opinion*$_1$ <b_1, d_1, u_1> and *issuer* field from the evidence statement E_1

Step2: Retrieve the evidence statement about issuer's testify_trust E_2<*I, issuer, testify_trust, opinion*$_2$> from local database

Step3: Extract *opinion*$_2$ <b_2, d_2, u_2> from the evidence statement E_2

Step4: Create a new evidence statement E_3<*I, subject, evidence, opinion*$_3$>. Compute *Opinion*$_3$<b_3, d_3, u_3 > by using the following formula. (*Discounting operator* defined in *Mathematical Theory of Evidence*)

(1) $b_3 = b_1 * b_2$

(2) $d_3 = b_1 * d_2$

(3) $u_3 = d_1 + u_1 + b_2 * u_1$

Step5: Compute probability expectation of < b_3, d_3, u_3 >

PE (*opinion*$_3$) = $b_3 + 0.5 * u_3$

Step6: RE (E_1) = PE (*opinion*$_3$)

5.2 Role assignment

Role assignment maps roles to users based on the evidence statements and role assignment policies. The research investigation in this component consists of:

1. Designing a role assignment policy declaration language.
2. Developing efficient algorithms to assign roles to users.

5.2.1 Policy declaration language

A role assignment policy is used to express the requirements for assigning a role to a user. The policy declaration language is used to specify:

- The content and the number of evidence statements needed for role assignment.
- A threshold value that characterizes the minimal reliability expected for each evidence statement. If the reliability associated with evidence does not meet the minimum threshold, this evidence will be ignored.

Currently, the declaration language supports limited data types and operators. We plan to extend the language and refine it by using XML to ease its portability in further research.

Syntax

```
Policy::= (PolicyDeclaration)*
PolicyDeclaration::=
                    role::=UnitDeclarations
UnitDeclarations::=  Unit ("^" Unit)*
Unit::=  "[" IssuerRole, EvidenceType, "{" Exp "}", Threshold, Redundency "]"
Threshold::=float
Redundency::=Integer
Exp::=  AndExp "||" Exp
AndExp::= OpExp "&&" AndExp
OpExp::= attr Op Constance
Constance::= integer | float | string
Op::= EQ | NEQ | GT | LT | EGT | ELT
```

A policy file can include several policy declarations. The name of a role is in Left hand of a policy declaration. The right hand of a policy declaration is unit declarations. Each unit declaration consists of one or more units. A unit is composed of *IssuerRole, EvidenceType, Exp, Threshold* and *Redundancy*. *IssuerRole* is the role a qualified issuer should hold. *EvidenceType* specifies the required evidence type. Conditions on the attributes of evidence are specified by using *Exp* that is an infix expression. The infix expression is transformed to a postfix expression and stored in local database to accelerate the evaluation, as illustrated by Figure 4. *Threshold* specifies the reliability of evidence. *Redundancy* is used to determine how many evidences satisfying above constraints are required.

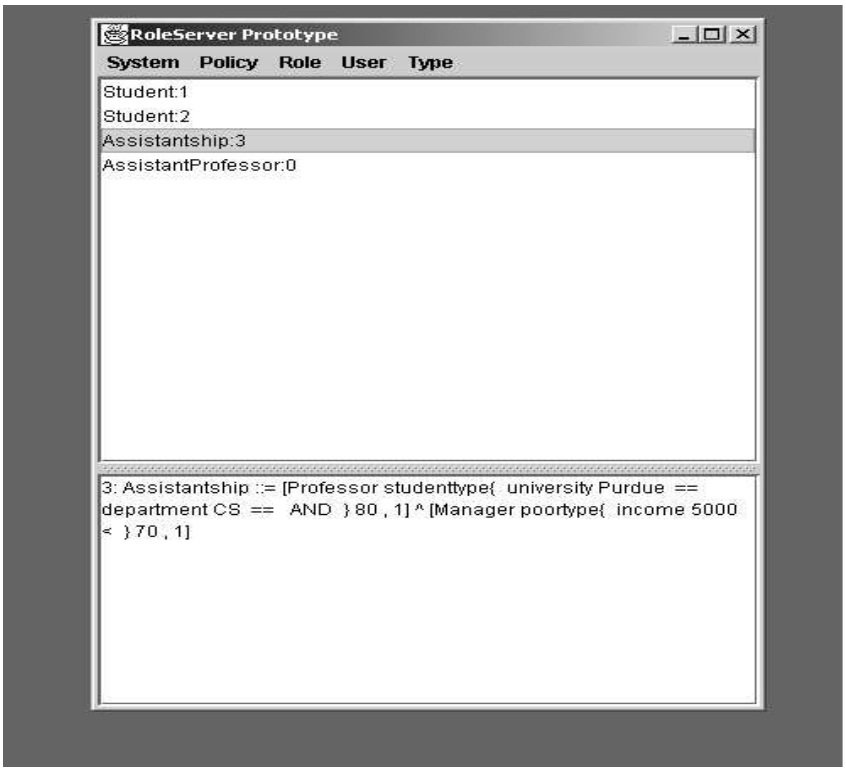

Figure 4 Internal Representation of a role-assignment policy

Example:

VIP::=[“Company”, “Manager”, {rank = “senior” && department = “sales” || salary > 100,000}, 75, 1] ^ [“I”, “access_trust”, {s>0.75 && c>0.5 && I>0.8}, 1, 1]

This policy specifies the conditions to get a VIP role. It consists of two units. The first unit requires that a user presents one evidence which says that he/she is a senior manager in sales department or his/her salary is greater than 100,000. The reliability of this evidence should not be lower than 75%. The second unit is the constraint on the user’s access_trust.

5.2.2 Evaluation policy

When a user presents a set of evidence, we need to determine a set of role-assignment policies that are satisfied with the given set of evidence. Several policies may be associated with a role. The role is assigned if and only if any of the policies is satisfied. A policy may contain several units. The policy is satisfied if and only if all units are evaluated TRUE.

Algorithm to assign a role

Input: a set of evidence *E* and their reliability, a role A

Output: True/False

P ← the set of policies whose left hand is role A
While P is not empty{
  q = a policy in P
  satisfy = true
  For each units *u* in q{
    If Evaluate_unit(*u*, *e*, *re*(*e*)) is false for all evidence statement *e* in *E*
      satisfy = false
  }
  If satisfy is true
    Return true
  Else
    Remove q from P

}
Return false

The algorithm to evaluate a unit is based on two assumptions: (1) the domains of attributes are infinite; (2) the distribution of attribute values is uniform.

Algorithm to evaluate a unit

Input: an evidence statement E_1 <*issuer, subject, evidence, opinion*$_1$> and its reliability RE (E_1), a unit of a policy *U*

Output: True/False

Step1: If *issuer* does not hold the *IssuerRole* specified in *U* or the type of *evidence* does not match *evidence_type* in *U*, return False.

Step2: Evaluate *Exp* of *U* as the following:

(1) If Exp_1 is "Exp_2 || Exp_3":

result(Exp_1) = max(result(Exp_2), result(Exp_3))

(2) If Exp_1 is "Exp_2 && Exp_3":

result(Exp_1) = min (result(Exp_2), result(Exp_3))

(3) If Exp_1 is "*attr* Op Constance c_1":

a. Op is *EQ*, *GT*, *LT*, *EGT*, *ELT*

   i. If *attr* OP Constance is true, result(Exp_1) = RE(E_1)

   ii. If *attr* OP Constance is false, result(Exp_1) = 0

b. Op is N^{EQ}:

   i. If *attr* OP Constance is true, result(Exp_1) = RE(E_1)

   ii. If *attr* OP Constance is false, result(Exp_1) = 1- RE(E_1)

Step3: If min (result (*Exp*), RE (E_1)) is greater or equal than *Threshold* in *U*, output True. Otherwise, output False

5.3 Trust information management

The process of assigning roles to users relies on trust information evaluated by the corresponding component. User/issuer trust is determined based on both direct-experience and recommendations. Unlike many research efforts that evaluate users' trust only based on recommendations, our trust information management consider both direct-experience and recommendations. The focus is on negative behavior that decreases the amount of trust the system has towards the user. The main functionality of trust information management includes mapping mistrust events to evidence statements and evaluating trust values.

Mapping mistrust events to evidence statements: A user's misbehavior is perceived by the system as **mistrust event** [15]. The component of trust information management maps mistrust events to evidence statements. Evidence statements provide an abstract view of the mistrust events to the trust evaluation model. First, mistrust events should be categorized. One category of mistrust events corresponds to an evidence type. Then, each category of mistrust events is represented by a set of characteristic features. However, different categorizes might have some common features. Our research investigation indicates that *criticality* and *lethality* proposed by Northcutt [19] can be used as such common features. Criticality measures the importance of the target of mistrust events. Lethality measures the degree of damage that could potentially be caused by mistrust events. The feature set of a category corresponds to the attribute set of an evidence type. Given a mistrust event, how to determine quantitative measures of its features is application-specific [5][21]. Finally, a mistrust event discovered by intrusion detection or data mining system is associated with a probability, which characterizes the confidence of the system

to make the claim. The probability is expressed by using the opinion parameter in evidence statement.

Evaluating Trust values: A user who first visits the system is assigned a trust value based on the default/average trust value of a trust environment that is similar as which the user is in. A trust environment consists of the role that the user requests, the domain/subnet from which the user comes, the trust opinion from third parties if available, and the trustworthiness of these third parties. For a known user, his/her trust value is adjusted mainly based on his/her behavior. Trust values are modified periodically. The access_trust values of users decrease if he/she involves in mistrust events. The testify_trust of a user u is modified periodically in the following way. Suppose u_1, u_2, ... u_n are assigned to access roles based on the evidences provided by u. The modification of testify_trust of u is related with the changes of access_trust of u_i ($1 \leq i \leq n$). The following example explains this idea. A professor has recommended ten students to a graduate school. If all students have poor academic performance, the recommendation letters from this professor becomes less convincing consequently.

5.4 Credential management

Credentials are available from local registry or provided by other service providers in the Internet in forms of certificates such as public key certificates, attributes certificates, etc. Since the user may not present all required credentials together with her request (e.g. when using SSL, only one certificate is sent from the subject to the server), automatic collection of missing credentials is required. Another important functionality of this component is to map different formats of credentials to evidence statements. The following issues need further research.

- Optimization of the process of credential collection: Will caching techniques optimize credential collection process? How will a pre-collection improve the response time? What are the network bandwidth and storage overheads introduced by pre-collection?
- The security issues related to the credential collection: How to establish secure communication channels among agents that collect credentials and the credential manger? How to protect the role server from denial-of-service attacks?

6. Conclusion

We present our research on authorization in this paper. We propose a trust-enhanced role-mapping server collaborating with a RBAC–enhanced web server to solve authorization in open environments. The role-sever determines a user's permission set based on trust and evidence. The representation and evaluation of evidence and trust are discussed. The algorithms of evaluation reliability of evidence and role-assignment policies are presented.

In addition to help solving authorization problem in open environments, this research can benefit trust and proof on the semantic web as well**.** The Semantic Web, which was thought up by Tim Berners-Lee, the inventor of the WWW, URIs, HTTP and HTML, is a mesh of information linked up in such a way as to be easily processable by machines on a global scale. An important principle is that "anything can say anything about anything". Thus, trust and proof plays a significant role in the Semantic Web. An ultimate goal of the Semantic Web is the capability of machine understanding and processing of information. The research on quantification of trust, formalization of evidence and trust, evaluation of reliability of evidences is an attempt for machine reasoning and proof. It will lead to an efficient way for determining the trustworthy of a piece of information on the Semantic Web.

Another area, which this research can be beneficial to, is decision-making in e-commerce**.** The question of trust and evidence is important in e-commerce. E-commerce is largely driven by database management software. Companies maintain huge amounts of user data and personal information about users. Misuse of such information even through authorized access should be denied. Trust is an important consideration. This research can thus be applied for effective trust management in E-commerce.

References

[1] A. Herzberg, Y. Mass, J. Mihaeli, D. Naor and Y. Ravid. Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers. *IEEE Symposium on Security and Privacy*, CA, 2000.

[2] A. Jøsang. A Logic for Uncertain Probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*. Vol. 9 No. 3, June 2001.

[3] A. Abdul-Rahman, S. Hailes. Supporting Trust in Virtual Communities. In *Proc. of Hawaii International Conference on System Sciences 33*, Maui, Hawaii, January 2000.

[4] An Internet Attribute Certificate Profile for Authorization: http://www.ietf.org/internet-drafts/draft-ietf-pkix-ac509prof-09.txt.

[5] D. Denning. *Information Warefare and Security*. Addison Wesley, 1999.

[6] D. McKnight and N. Chervany. Conceptualizing Trust: A Typology and ECommerce Customer Relation Model. In *Proc. of the 34th Hawaii ICSS-2001*, 2001.

[7] E. Bertino, E. Ferrari and E. Pitoura. An Access Control Mechanism for Large Scale Data Dissemination Systems. *RIDE-DM 2001*, 2001.

[8] G. Ahn and R. Sandhu. Role-based Authorization Constraints Specification. *ACM Transactions on Information and System Security*, Vol. 3, No. 4, Nov. 2000.

[9] G. Shafer. *A Mathematical Theory of Evidence*. Princeton University Press, 1976.

[10] Internet X.509 Public Key Infrastructure Certificate and CRL Profile. http://www.ietf.org/rfc/rfc2459.txt.

[11] J. Joshi, W. Aref, A. Ghafoor and E. Spafford. Security Models for Web-Based Applications. *Communications of the ACM*, 44(2):38-44, 2000.

[12] J. Park and R. Sandhu. Role-based Access Control on the Web. *ACM Transactions on Information and System Security*, Vol. 4, No. 1, Feb. 2001.

[13] M. Blaze, J. Feigenbaum and J. Lacy. Decentralized Trust Management. In *Proc. of the 17th Symposium on Security and Privacy*, 1996.

[14] M. Blaze, J. Feigenbaum, J. Ioannidis and A. Keromytis. The KeyNote Trust-Management System. http://www.cis.upenn.edu/~angelos/keynote.html.

[15] M. Mahoui, B. Bhargava and Y. Zhong. Separating Between Trust and Access Control Policies: A Necessity for Web Applications. In *Proc. of the IEEE Workshop on Security in Distributed Data Warehousing*, New Orleans, 2001.

[16] M. Winslett, N. Ching, V. Jones and I. Slepchin. Using Digital Credentials on the World-Wide Web. *Journal of Computer Security*, 1997.

[17] P. Resnick and J. Miller. PICS: Internet Access Controls without Censorship. *Communications of the ACM*, Vol. 39, No. 10, 1996.

[18] R. Khare and A. Rifkin. Weaving a Web of Trust. *World Wide Web Journal, special issue on security*, Vol. 2, No. 3, Summer 1997.

[19] S. Northcutt. *Intrusion Detection: Analyst's Handbook*. New Riders, 1999.

[20] S. Marsh. Formalizing Trust as a Computational Concept. Ph.D. Thesis. University of Stirling, 1994.

[21] W. Lee, W. Fan, M. Miller, S. Stolfo, F. Zadok. Toward Cost-Sensitive Modeling for Intrusion Detection and Response. *Journal of Computer Security*, 2001.

[22] Y. H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick and M. Strauss. REFEREE: Trust Management for Web Applications. *Word Wide Web Journal*, pp. 127-139, 1997.