CERIAS Tech Report 2002-21

AUTHORIZATION BASED ON EVIDENCE AND TRUST

by Bharat Bhargava and Yuhui Zhong

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

CERIAS Tech Report 2002-21

Authorization Based on Evidence and Trust [*]

Bharat Bhargava and Yuhui Zhong

Center for Education and Research in Information Assurance and Security (CERIAS),
and Department of Computer Sciences
Purdue University, West Lafayette, IN 47906-1398, USA {bb, zhong}@cs.purdue.edu

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 her objective properties that are demonstrated by digital credentials (evidences). However, holding credentials is not sufficient to certify that a user is trustworthy. Therefore, we propose using the notion of trust to characterize the probability that a user will not harm an information system. We present a trust-enhanced role-mapping server, which cooperates with RBAC (Role-Based Access Control) mechanisms to together implement authorization based on evidence and trust. A prerequisite for this is our proposed formalization of trust and evidence.

1 Introduction

Research is needed to develop authorization mechanisms for a large and open community of users. In such an environment, prior knowledge about a new user normally does not exist [20]. For authorization, the permission set for each user must be determined. Current research efforts grant privilege to a user based on her objective properties that are demonstrated by digital credentials (evidences) issued by third parties [4],[9]. Credentials are not sufficient to certify that a user is trustworthy. Therefore, a formalized notion of trust is used by us to characterize the probability that a user or an issuer of credentials will not carry out harmful actions [6]. Next, the impact of users' behavior on system's trust towards them needs to be quantified. Furthermore, the reliability of evidence or credentials from different issuers might be different. Authorization based on evidence as well as trust makes access control adaptable to users' or issuers' behavior. The research requires: (1) an appropriate representations of the evidence and trust, so that their manipulation can be automated, (2) a suitable authorization architecture that can incorporate the evidence and trust, and (3) integration of this scheme with existing access control mechanisms. We investigate these issues and propose a trust-enhanced role-mapping (TERM) server architecture,

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

which can cooperate with RBAC (Role-Based Access Control) mechanisms for authorization based on evidence and trust.

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 a TERM 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 that evaluate the reliability of evidence and role-assignment policies. Conclusions are in section 6.

2 Related Work

Authorization in an open environment: This is an active area of research. One direction is *trust management* [4],[5]. 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 the existing ones such as RBAC [9].

Another direction of research divides the authorization problem into two subproblems: (1) determine the permission set of a user (2) enforce access control by using existing mechanisms like RBAC. These approaches have the advantage of easy integration with existing systems. Our research effort is in this direction. Others determine users' permission set only according to evidence/credentials. Our work is distinguished by using evidence and trust.

Trust Models: Several researchers have proposed algorithms to summarize trust opinions from third parties. The summarization includes evaluating an opinion from an issuer, or combining opinions from different issuers [1],[11],[14]. Little research has 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 this first-hand information in our framework.

RBAC: RBAC has emerged as a promising technology for efficiently managing and enforcing security in large organizations [2],[17]. 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 sets. User-role mapping assigns roles to users.

3 Concepts and Formal Definitions

The following concepts, definitions and representations are used in our research.

3.1 Concepts

Evidence: Evidences (also called credentials) are statements about certain properties of an entity (called subject) . An evidence can come from internal or external sources. Evidence can be information stored in a local database (e.g, user name and password) or public key certificate (e.g, X.509 V3) [8],[10], digitally signed document (e.g, PICS rating) [18], etc.

Issuer's opinion about evidence: 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. An issuer's opinion about an evidence characterizes the degree to which the issuer is sure about the statement he/she makes.

Reliability of evidence: The reliability of an evidence represents the subjective degree of belief in the evidence of the entity relying on the evidence. The reliability of an evidence depends on issuer's opinion and relying party's opinion about the issuer.

Trust towards a user or an issuer: Trust is a subjective degree of belief [15] in harmlessness of a user. The aspects forming the trust and the weights of the aspects might be different for different entities (users or issuers), or for a given entity in different environments.

Direct experience and recommendation: The interactions between the observer and the observed entity are called "direct experience", and are first-hand information. The opinions about an entity obtained from other entities are called "recommendations," and are second-hand information. Because trust is not transitive [1], recommendations cannot be directly used. Trust opinion is formed mainly based on direct experience and, to a lesser degree, on recommendations.

Trust associated with an issuer and with a regular user: Trust associated with an issuer should be distinguished from one associated with a regular user. The former impacts the trust towards the evidence provided by the issuer. The latter characterizes the trust towards the user's own behavior.

Trust environment: Trust is environment-specific [15]. Different aspects of trust might be emphasized in different environments. The measurement of the same aspect of trust may vary in different environments. Representing an environment and propagating trust in different environments are 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 $\in$ {opt, mand} specifies whether the attribute type is optional or mandatory. Evidence type specifies information that is required by different 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: *credentials_evidence* and *trust_evidence*. Credentials_evidence includes the set of all possible credential types recognized by the role server. Trust_evidence includes the set of all possible trust types used by the TERM server to describe trustworthiness. Level 2 consists of access_credentials, access_trust, testify_credentials, and testify_trust. Access_credentials and access_trust represent credential/trust related to regular user. Testify_credentials and testify_trust are used to represent credential/trust related to an issuer. The remaining evidence types inherit properties of one of the four Level 2 evidence types.

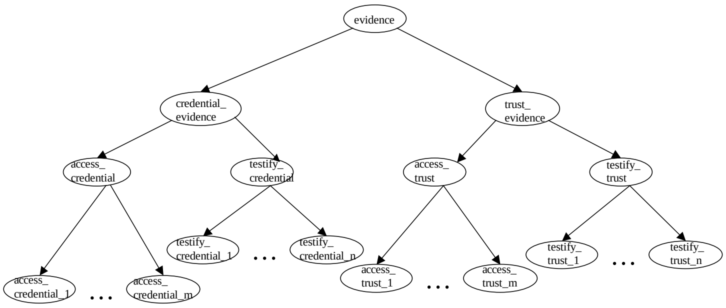

Fig. 1. Evidence Hierarchy

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* = (a_1:v_1,. . . , a_n:v_n), where a_1, . . . , a_n are the names of attributes, v_1, . . . , v_n are their values. Evidence is an instance of an evidence type. (Cf. the credentials model [3].)

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.

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 ∈ [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 here that uncertainty about belief and disbelief can be split equally between them based on the principle of insufficient reason [19].

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. *Opinion* characterizes the issuers belief related to the *evidence.*

An evidence statement provides a uniform view of different kinds of credential and trust information. It associates credentials or trust with different belief degree (expressed by an opinion), and makes it easy to adopt new type of credentials.

Role classification: Without a loss of generality, roles are classified into two non-overlapping categories.

Access role: A role is an *access role* if its permission set includes particular types of access to one or more objects of the system. A regular user must hold certain access roles.

Testifying role: A role is a *testifying role* if its permission set includes providing evidence for other entities. An issuer must hold certain testifying roles. The system accepts the evidence only from issuers holding appropriate testifying roles specified in the mapping policies.

Representation of trust information: Evidence statements are used to convey trust information.

Trust related to access roles: Trust for access roles is represented as (*I*, *u*, *access_trust*, *opinion*). *I* denotes the TERM server itself, *u* refers to the user, *opinion* denotes how much TERM server believes the above statement, and *access_trust* is an evidence type, which shows trust that the user will not harm the system. It contains three attributes (*ua*, *mc*, *il*), with the domains [0, 1]. Each attribute characterizes one aspect of user's potential harmful actions. The higher the value, the higher the probability that a user will not carry out such harmful actions.

1. Attribute *ua* denotes trust that the user will not attempt to get unauthorized access.

2. Attribute *mc* characterizes trust that the user will not try malicious consumption of enormous amounts of resources.

3. Attribute *il* shows a belief that the user will not try to cause an information leak.

Trust related to testifying roles: Trustworthiness for testifying role is represented as (*I*, *u*, *testify_trust*, *opinion*). *I*, *u* and *opinion* are the same as above. *Testify_trust* is an evidence type, which shows trust that the user will provide accurate information about other users. *Testify_trust* contains one attribute (*t*) with the domain [0, 1]. The higher the value, the higher belief that an evidence provided by the corresponding user is trustworthy.

4 Architecture of TERM server

The proposed TERM server collaborates with an RBAC-enhanced web server for authorization in open environments. The task of the TERM server is to map users to roles based on evidence and trust. Clients obtain the roles from a TERM 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 the appropriate roles, and sends back the object if the answer is true. The focus of this paper is on the TERM server.

The TERM server first collects credentials and transforms them to evidence statements. Then, it evaluates the reliability of the evidence based on the *opin-*

ion attribute of the evidence statement, and on testify_trust related to issuer. Finally it maps users to roles based on assignment policies, evidence reliability, and users trustworthiness. The top-level view of the architecture of a TERM server is shown in Figure 2. There are four components that exchange information using evidence statements presented. *Credentials Management* transforms different formats of credentials to evidence statements. *Evidence Evaluation* evaluates the reliability of evidence statements. *Role Assignment* maps roles to users based on the evidence statements and role assignment policies. *Trust Information Management* evaluates user/issuers trust information based on system's direct experience and issuers' recommendations.

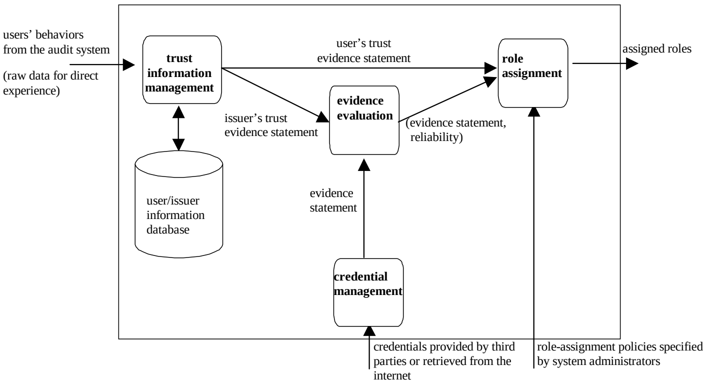

Fig. 2. Architecture of a TERM server

5 Algorithm and implementation

A role-assignment policy declaration language has been designed 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. A prototype including *Evidence Evaluation* and *Role Assignment* and a part of *Trust Information Management* has been implemented (the *Credentials Management* component is still under development).

5.1 Evidence Evaluation

The *Evidence Evaluation* component determines the reliability of evidences for the TERM server. The reliability is computed on the basis of the opinions included in the evidence statements, and the issuers *testify_trust*. The ratio of belief to disbelief may affect the distribution of uncertainty. We plan to investigate this topic in our 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: get opinion$_1$ = (b_1, d_1, u_1) and *issuer* attribute from E_1
Step2: get testify_trust of *issuer*: E_2 = (I, *issuer*, *testify_trust*, $opinion_2$) from the local database
Step3: 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 formulas (the discounting operator is defined in [19]):

$$b_3 = b_1 * b_2,\ d_3 = b_1 * d_2,\ u_3 = d_1 + u_1 + b_2 * u_1$$

Step4: The *probability expectation* for (b_3, d_3, u_3) gives us the reliability for E_1, hence RE (E_1) = $b_3 + 0.5 * u_3$

5.2 Role assignment

For this component, we devised a role-assignment policy declaration language, and developed algorithms to assign roles to users.

Policy declaration language: The policy declaration language is used to specify: (1) the content and the number of evidence statements needed for role assignment; (2) a threshold value that characterizes the minimal reliability expected for each evidence statement. If the reliability associated with an evidence does not exceed the minimum threshold, this evidence will be ignored.

Syntax
Policy::= (PolicyDeclaration)*
PolicyDeclaration::= Role_Name = UnitDeclarations
Role_Name::=*string*
UnitDeclarations::=Unit ("$\bigwedge$" Unit)*
Unit::= "[" *IssuerRole*, *EvidenceType*, "{" Exp "}", Threshold, Nr_Stmts "]"
Threshold::=*float*
Nr_Stmts::=*integer*
Exp::= AndExp "||" Exp
AndExp::= OpExp "&&" AndExp
OpExp::= *attr* Op Constant
Constant::= *integer* | *float* | *string*
Op::= $= \mid \neq \mid > \mid < \mid \geq \mid \leq$

A policy file can include several policy declarations. The name of a role is on the left hand side of a policy declaration. The right hand side of a policy declaration includes unit declarations. Each *UnitDeclarations* consists of one or more *Units*. A *Unit* is composed of *IssuerRole*, *EvidenceType*, *Exp*, *Threshold* and *Nr_Stmts*. *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*. *Threshold* specifies the minimum required value for the reliability of evidence. *Nr_Stmts* is used to determine how many evidences satisfying the above conditions are needed.

Example: VIP::=["Company", "Manager", {rank = "senior" && department = "sales" || salary > 100,000}, 0.75, 1] $\bigwedge$ ["I", "access_trust", {ua>0.75 && mc>0.5 && il>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 she is a senior manager in sales department, or 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.

Evaluation policy: When a user presents a set of evidences, we need to determine a set of role-assignment policies that are satisfied by this set of evidences. 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 its units evaluate to *True*.

Algorithm to assign a role to a user

Input: a set of evidences E with their reliabilities for a user, a role R

Output: *True*/*False*

```
P is the set of policies with role R on their left hand side
while P is not empty
    p = a policy in P
    satisfy = True
    for each unit $u$ in p
        if Evaluate_unit($u$, $e$, $RE(e)$) is False for all evidence statements e in E
            then satisfy = False
    if satisfy = True then return True else remove p 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 of a role-assignment policy

Input: an evidence statement $E_1 = (issuer, subject, evidence, opinion_1)$ and its reliability RE (E_1), a unit U of a policy

Output: *True*/*False*

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

Step2: Evaluate each *Exp* of U as follows:

```
if $Exp$ = "$Exp_1$ || $Exp_2$" then result($Exp$) = max(result($Exp_1$), result($Exp_2$))
else if $Exp$ = "$Exp_1$ && $Exp_2$" then result($Exp$) = min (result($Exp_1$), result($Exp_2$))
else if $Exp$ = "$attr$ Op Constant" then
    if $attr$ OP Constant = True then result($Exp$) = RE($E_1$)
    else if OP = $\neq$ then result($Exp$) = 1 - RE($E_1$)
    else result($Exp$) = 0
```

Step3: if min (result(*Exp*), RE (E_1)) $\geq$ *Threshold* in unit U, output *True*. Otherwise, output *False*.

5.3 Trust information management

The trust information management component executes two important steps. First, it maps mistrust events to evidence statements, and then it appropriately updates trust values in the user/issuer trust information database.
Mapping mistrust events to evidence statements: A user's misbehavior is perceived by the system as a *mistrust event* [13]. Mistrust events are categorized. One category of mistrust events corresponds to one evidence type. Each category of mistrust events is represented by a set of characteristic features. The feature set of a category corresponds to the attribute set of an evidence type. Different mistrust event categories might have some common features. For example, *criticality* and *lethality*[16] 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. Given a mistrust event, how to determine quantitative measures of its features is application-specific [7][12]. A mistrust event discovered by intrusion detection or data mining (both are external to our TERM server) is associated with a probability provided by them. This probability characterizes the confidence that a user caused a harm to the system. The probability impacts the opinion parameter in the evidence statement.
Updating trust values in the user/issuer trust information database: A user who visits the system for the first time is assigned a trust value based on the default/average trust value of her trust environment or a similar one. 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. With time the user becomes known to the system. Now her trust value is adjusted mainly based on her behavior. Trust values are modified periodically. The *access_trust* values of a user decrease if she was involved in any mistrust event. The *testify_trust* of a user u is modified periodically in the following way. Suppose u_1, u_2, $\ldots$, u_n are assigned to access roles based on the evidences provided by u. The modification of *testify_trust* of u is related to the changes of *access_trust* of all u_i's.

6 Conclusions

In this paper, we propose a detailed architecture for a TERM server. This server collaborates with an RBAC-enhanced web server to solve the authorization problem in open environments. The TERM server determines a user's permission set based on trust and evidence. Representations for evidence and trust, and evaluation of both of them are discussed. The algorithms for evaluation of evidence reliability and for role-assignment policies are presented.

In addition to showing our authorization solution, our result can contribute to solving the issues of trust and proof on the semantic web. An ultimate goal for the semantic web research is gaining the capability of machine understanding of information. Our research on quantification and formalization of evidence and trust could also help to enhance machine reasoning and proof. It could lead to an efficient way for determining trustworthiness of information on the semantic web. Another area, which could benefit from our research is decision-making in e-commerce, especially in effective trust management. Misuse of company information even through authorized access should be denied, therefore the question of trust and evidence is extremely important.

References

1. A. Abdul-Rahman and S. Hailes. Supporting trust in virtual communities. In *Hawaii International Conference on System Sciences*, Hawaii, January 2000.
2. G. Ahn and R. Sandhu. Role-based authorization constraints specification. *ACM Transactions on Information and System Security*, 3(4), November 2000.
3. E. Bertino, E. Ferrari, and E. Pitoura. An access control mechanism for large scale data dissemination systems. In *RIDE-DM 2001*, 2001.
4. M. Blaze, J. Feigenbaum, and J. Ioannidis. The keynote trust-management system version 2, http://www.ietf.org/rfc/rfc2704.txt.
5. M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized trust management. In *the 17th Symposium on Security and Privacy*, 1996.
6. Y. H. Chu, J. Feigenbaum, B. LaMacchia, P. Resnick, and M. Strauss. Referee: Trust management for web applications. *Word Wide Web Journal*, 1997.
7. D. Denning. *Information Warefare and Security.* Addison Wesley, 1999.
8. S. Farrell and R. Housley. An internet attribute certificate profile for authorization, http://www.ietf.org/internet-drafts/draft-ietf-pkix-ac509prof-09.txt.
9. A. Herzberg, Y. Mass, J. Mihaeli, D. Naor, and Y. Ravid. Access control meets public key infrastructure, or: Assigning roles to strangers. In *IEEE Symposium on Security and Privacy*, CA, 2000.
10. R. Housley, W. Ford, W. Polk, and D. Solo. Internet x.509 public key infrastructure certificate and crl profile, http://www.ietf.org/rfc/rfc2459.txt.
11. A. Jsang. A logic for uncertain probabilities. *International Journal of Uncertainty, Fuzziness and Knowledge-based Systems*, 9(3), June 2001.
12. W. Lee, W. Fan, M. Miller, S. Stolfo, and F. Zadok. Toward cost-sensitive modeling for intrusion detection and response. *Journal of Computer Security*, 2001.
13. M. Mahoui, B. Bhargava, and Y. Zhong. Separating between trust and access control policies: A necessity for web applications. In *the IEEE Workshop on Security in Distributed Data Warehousing*, New Orleans, 2001.
14. S. Marsh. *Formalizing Trust as a Computational Concept.* PhD thesis, University of Stirling, UK, 1994.
15. D. McKnight and N. Chervany. Conceptualizing trust: A typology and ecommerce customer relation model. In *the 34th Hawaii ICSS-2001*, Hawaii, 2001.
16. S. Northcutt, J. Novak, and D. McLachlan. *Network Intrusion Dectection: Analyst's Handbook.* New Riders Publishing, 1999.
17. J. Park and R. Sandhu. Role-based access control on the web. *ACM Transactions on Information and System Security*, 4(1), February 2001.
18. P. Resnick and J. Miller. Pics: Internet access controls without censorship. *Communications of the ACM*, 39(10), 1996.
19. G. Shafer. *A Mathematical Theory of Evidence.* Princeton University Press, 1976.
20. M. Winslett, N. Ching, V. Jones, and I. Slepchin. Using digital credentials on the world-wide web. *Journal of Computer Security*, 1997.