Available online at www.sciencedirect.com

ScienceDirect

Electronic Notes in Theoretical Computer Science 197 (2008) 169–177

Electronic Notes in Theoretical Computer Science

www.elsevier.com/locate/entcs

A Trust- and Property-based Access Control Model

Joachim Biskup[1] Julia Hielscher[2,4] Sandra Wortmann[3]

Information Systems and Security
University of Dortmund
Germany

Abstract

In open and distributed property-based access control systems, access rights are granted because of presented certified properties. However, controlling agents base their access decisions not only on presented certified properties. Crucial factors are the correlations between certified properties, experiences and expectations concerning other participating agents, and the resulting trust modalities. This work identifies the essential correlations and demonstrates how they can be explicitly integrated in access decisions.

Keywords: property-based access control, trust modalities

1 Introduction

Open and distributed IT-systems require elaborate and dynamic ways to enforce access control. As we focus on distributed systems with autonomously acting agents, traditional global or hierarchical access control systems are not suitable. In such distributed IT-systems, each agent controlling a resource needs to discretionarily and dynamically maintain and enforce the resource's access control policy. Furthermore, requesting agents are usually not identified or registered. For such a setting, attribute-based access control models are well suited, see e.g. [1,4,12,13,16]. In this work, we prefer using the term *property-based access control* model, which originates from a later on introduced important distinction between different types of a requesting agent's property. Thus, each controlling agent states his resource's access control policy in terms of properties that are required for accessing the corresponding resource.

[1] Email: Joachim.Biskup@uni-dortmund.de

[2] Email: Julia.Hielscher@uni-dortmund.de

[3] Email: Sandra.Wortmann@uni-dortmund.de

[4] current affiliation: Software System Engineering, ICB, University of Duisburg-Essen, Germany

doi:10.1016/j.entcs.2007.12.026

There exist two main issues that need to be considered in property-based access control systems. First, a controlling agent need to be aware of the "property scheme" which is used by a certifying agent. Otherwise properties that are encoded in submitted certificates do not necessarily match those requested by the corresponding access control policy. Second, existing approaches for property-based access control systems are usually monotonic in the sense that more certified properties imply more positive access decisions.

To face the first issue, we deploy a comprehensive view of trust. Instead of assuming a global property scheme, controlling agents decide on the trustworthiness concerning certifying agents. As earlier proposed in [15], each access control policy reflects (local) trust modalities of the respective controlling agent. It turns out that local trust modalities are determined by different local and global aspects of the underlying system. To face the second issue, our model considers revocation of already certified properties, explicit access prohibitions and statements about trust and distrust concerning other agents.

We conceptually develop the fundamentals in Section 2 by analysing the correlation of agents' properties, expectations and resulting trust modalities with respect to decisions on property conversion and (direct) access requests. We suggest credential-based implementation ideas in Section 3. In Section 4 we identify so-called *structures* that our model consists of: the *issuing structure*, the *forwarding structure* and the *trust structures*. In Section 5 we give an example for defining security policy rules, and we conclude in Section 6.

2 The Trust- and Property-based Fundamentals

Hereafter, we use the terms free and bound property as introduced in [3]. Assigning participants certify *free properties* without any further intention on their usage: Free properties are independent of any resource or access decision. They can have positive, negative, or neutral influences on access decisions of a controlling agent. A resource's owner certifies *bound properties* in terms of a specific resource: Bound properties are regarded as permissions to use the resource. For getting a bound property, a requester has to present requested free properties. The granting of a bound property caused by the verification of presented free properties is called *property conversion*. For direct access, a requester has to present the corresponding bound property to the controlling participant.

It is important to note that this distinction relates to an owner-specific view. Bound properties that are granted within an owner's domain may be regarded as free properties within another owner's domain.

Considering a monotonic access control system, an owner can express only *trust* in other participants: He trusts certain assigning participants to correctly assign free properties, and he trusts the holders of these certified free properties to harmlessly access his resource. However, such a monotonic access control system is too restrictive. In simple non-monotonic access control systems, access rights are granted because of decisions that base on positive experiences and expectations, whereas negative experiences and expectations lead to removing existing access rights.

We specifically argue for dealing with *doubt* and *distrust* as well. Therefore, we extend the meaning of bound properties as follows. Trust on the side of an owner leads to granting a bound property that is regarded as a positive access right. Distrust leads to granting a bound property that is regarded as a negative access right. Thus, distrust does *not* lead to removing a granted positive access right. Whenever an owner has doubts on a granted bound property, he removes it.

2.1 Definitions of Knowledge and Local Evaluation

First, we define *trust modalities* in the form of relationships between two participants of the computing environment. As we focus on an access control system, we consider how the owner's trust modalities are determined and further how they influence his decisions on property conversion. The owner trusts, distrusts or has doubts on assigners, he is aware of, concerning their tasks. This means that an owner can take on several trust modalities concerning one assigner, namely one trust modality concerning each property that is certified or is announced to be certified by the respective assigner. The owner's trust modalities influence his decisions on property conversion by being used in his security policy as shown in Section 5 with an example. There is a growing variety of meanings of trust (modalities), see e.g., [11,6,13,8]. As a minimal agreement, trust might be defined as "a firm belief in the competence of an entity to act dependably, [. . .], within a specified context" [8]. We want to explore the origin of belief and add distrust and doubt. Therefore, we define the trust modalities on the basis of a participant's knowledge.

Let sets $\mathcal{A}$ and $\mathcal{P}$ be given that denote the participating agents and properties, respectively. The *global knowledge* $\mathcal{K}$ is defined as $\mathcal{K} \subset \mathcal{A} \times \mathcal{P}$. The global knowledge is the set of grantor-property pairs

(i) that deal with free or bound properties that are actually certified by the respective granting participant, or

(ii) for which the stated granting participant has publicly announced that he will certify the respective property.

As trust modalities always refer to a particular participant, we define the *local knowledge* $\mathcal{K}_a$ of participant a as $\mathcal{K}_a \subseteq \mathcal{K}$. This means that the local knowledge $\mathcal{K}_a$ of a participant $a \in \mathcal{A}$ is the collection of all grantor-property pairs that the participant *is aware of* [5] . For determining his trust modalities, each participant evaluates his local knowledge restricted to those grantor-property pairs that deal with free properties from his specific view. By abuse of language, we hereafter identify the local knowledge $\mathcal{K}_a$ of participant a with the set of grantor-property pairs containing free properties from the participant-specific view. Then, $\mathcal{K}_a$ is evaluated by a local function $eval_a$ that maps a grantor-property pair (a_i, p_i) with $a_i \in \mathcal{A},\ p_i \in \mathcal{P}$ to a trust modality $t \in \mathcal{T}$ with $\mathcal{T} := \{TRUST, DOUBT, DISTRUST\}$. Thus, the local evaluation function $eval_a$ of participant a is defined as $eval_a : \mathcal{K}_a \rightarrow \mathcal{T}$. The eval-

[5] The meaning of *being aware of* is explained in Section 4.

uation is determined by *expectations* of the corresponding participant a as follows. If the participant expects that for a given grantor-property pair (a_i, p_i) assigner a_i honestly certifies or will honestly certify property p_i, then a *positive evaluation* maps the grantor-property pair to trust, i.e. , $eval_a(a_i, p_i) = TRUST$. Analogously, a *negative evaluation* maps a given grantor-property pair to distrust, and a *neutral evaluation* maps a given grantor-property pair to doubt. Obviously, the local evaluation function leads to the disjoint *classification sets* $\mathcal{K}_a^{TRUST}$, $\mathcal{K}_a^{DISTRUST}$, and $\mathcal{K}_a^{DOUBT}$ with

$$\mathcal{K}_a = \mathcal{K}_a^{TRUST} \dot\cup \mathcal{K}_a^{DISTRUST} \dot\cup \mathcal{K}_a^{DOUBT}.$$

The classification sets represent the participant's trust modalities.

2.2 Expansion of Local Knowledge and Refreshed Expectations

Any new participant of the computing environment starts with *initial experiences* coming from outside the system. Caused by his experiences, the participant forms his expectations concerning the believed behaviour of other participants. Then, the participant evaluates his local knowledge as introduced above and gains the classification sets. Any new grantor-property pair the participant gets informed of leads to an expansion of his local knowledge[6] . We consider two cases:

(i) A *"redundant" expansion of local knowledge* occurs, if the owner gets informed about a grantor-property pair that is already mapped to one of his classification sets. The current evaluation of his local knowledge already considers the grantor-property pair and no new evaluation of his local knowledge is necessary.

(ii) An *"effective" expansion of local knowledge* occurs, if the owner gets informed about a so far unknown grantor-property pair. In this case, a new evaluation of the now enlarged local knowledge is necessary. The classification sets remain unaltered, except for the new mapping of the grantor-property pair.

Refreshed expectations are caused by the owner's experiences, e.g. , when a requester acts against the owner's expectations. In this case, the owner's experiences change. Consequently, he refreshes his expectations and the local evaluation is reperformed. This might lead to an alteration of the classification sets.

3 A Certificate- and Credential-based Implementation

For certifying properties it is common practice to use certificates. The most well-known public key infrastructures include X.509 [9] and SPKI/SDSI [7]. Following [3], we use the terms *certificates* and *credentials* as follows: Certificates are used to certify free properties whereas credentials are used to certify bound properties. Further, we postulate that for each property $p \in \mathcal{P}$ there exists an "opposite property" $\complement p \in \mathcal{P}$.

We consider two kinds of access requests to a particular resource:

[6] How a participant gets informed of new grantor-property pairs is described in Section 4.

(i) The requester suitably presents collected attribute certificates in order to gain a specific authorisation credential. Based on his security policy rules, the owner decides on the *request for property conversion*. In the case that the owner will grant a bound property, we distinguish between a successful and an unsuccessful property conversion. The distinction depends on whether the bound property is regarded as a positive access right or as a negative access right, respectively. A successful property conversion is materialized by issuing an authorisation credential, and an unsuccessful property conversion is materialized by issuing a prohibition credential. In the case that the owner's verification of the presented attribute certificates fails, his decision leads to a *failed property conversion*: The request is rejected and no credential is issued.

(ii) The requester presents an authorisation credential for the respective resource by submitting a collected authorisation credential or by immediately presenting the result of a successful property conversion. Thus, no property conversion is necessary for a *direct access request*. The owner decides on the (direct) access request by simply verifying the presented authorisation credential.

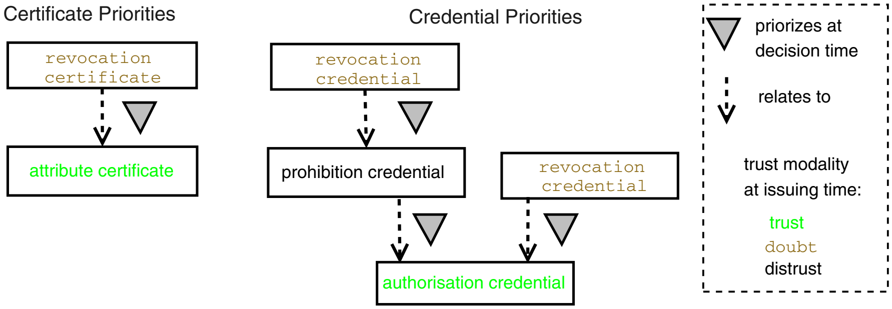

Fig. 1. Certificate and Credential Priorities at Decision Time

Summarizing, Figure 1 depicts the priorities of certificates and credentials, assuming the owner is aware of them at decision time. For placing confidentiality and integrity over availability of the resource, we priorize a prohibition credential over the corresponding authorisation credential. The declared priorities should be considered as defaults; in special situations different priorities might be reasonable.

We stress that we do not postulate the use of a particular public key infrastructure. An emerging query-and-response protocol standard for exchanging authorisation and authentication information is SAML [5]. Depending on the underlying infrastructure one can also employ the SAML protocol for exchanging participants' properties. Considering our scenario, the most relevant SAML statements are *attribute assertions* by which an assertion subject is associated with stated attributes. SAML attribute assertions may be used as input to authorisation decisions made according to the XACML standard [14]. Among an authorisation architecture proposal, XACML defines an XML-dialect for specifying attribute-based access control policies. In case of using XACML policies, the exchange of requested properties can be done via both certificates/credentials and SAML.

4 Structures of the Model

The presented conceptual model bases on three essential structures: (1) The *issuing structure*, (2) the *information forwarding structure*, and (3) the *trust structures.* Each structure encloses different aspects that are produced due to participants' tasks or other actions such as announcing grantor-property pairs or evaluating local knowledge. The issuing structure and the information forwarding structure are *global structures*, because the enclosed aspects are produced by the tasks and actions of all participants of the computing environment. Even so, the information forwarding structure produces local aspects for particular participants. Further, each participant maintains a completely *local structure*, namely his own trust structure. The structures are cyclicly dependent from each other as visualized in Figure 2.

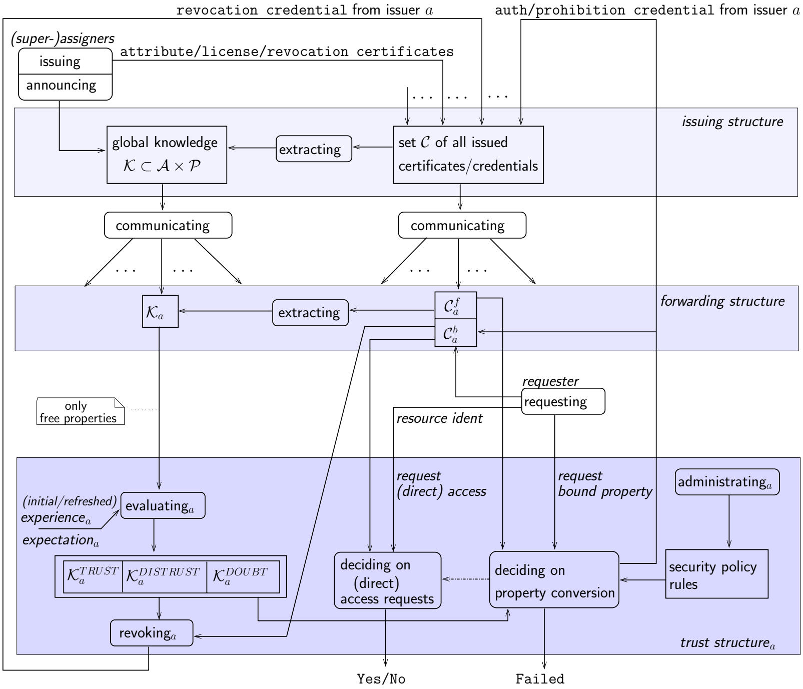

Fig. 2. Cycle of the Structures

The *issuing structure*, depicted at the top of the cycle, encloses the set $\mathcal{C}$ of all issued certificates and credentials and the global knowledge $\mathcal{K}$. The global knowledge is produced by extracting the grantor-property pairs from each attribute certificate and authorisation credential in $\mathcal{C}$ and by assigners' announcements. Assigners may publicly announce that they will issue particular grantor-property pairs.

The *information forwarding structure* encloses the local knowledge $\mathcal{K}_a$ of each participant a and the local set of certificates $\mathcal{C}_a^f$ and the local set of credentials $\mathcal{C}_a^b$ the participant a is aware of. These aspects are produced by communicating actions of the participants. Each participant is aware of credentials and certificates he has issued and additionally of those granted to him. The local knowledge of a participant is produced by extracting the grantor-property pairs from each attribute certificate in the local set of certificates and additionally by other communicating actions such as getting informed of public announcements.

As introduced in Section 2, each participant determines his trust modalities by evaluating his local knowledge about grantor-property pairs that deal with free properties. The local evaluation depends on the participant's (initial or refreshed) expectations. Each local *trust structure*$_a$ encloses the classification sets of the respective participant a. A refreshed experience may lead to an alteration of the classification sets. Depending on already issued credentials and the security policy rules, the altered classification sets may cause the issuing of revocation credentials.

The enclosed security policy rules are administrated by the respective participant a. Initiated by a request action, participant a decides on property conversion depending on the classification sets, the security policy rules and the local set of certificates $\mathcal{C}^f_a$. Decisions on property conversion influence the issuing structure as follows: For each (successful and unsuccessful) property conversion, participant a issues a corresponding authorisation or prohibition credential. It is possible to immediately decide on a (direct) access request after a (un)successful property conversion.

5 An Example for Security Policy Rules

An owner may explicitly define security policy rules in order to take decisions on property conversion. Here, we exemplarily sketch a simple security policy specification language that can be easily integrated in logic-based frameworks for flexible authorization mechanisms like, e.g., [2,10]. Our first-order logic language uses the following vocabulary: constants for resources, participating agents, and properties, and variables for participating agents and properties; the predicate symbols `trust`, `doubt`, and `distrust` for classification sets, and the predicate symbol `cert` for certified free properties; the predicate symbols `suc` and `unsuc` for successful and unsuccessful property conversions, respectively. Further, Boolean connectives and the common quantifiers are used. A *security policy rule* is a rule of one of the forms: (1) $\texttt{suc}(r, a) \leftarrow \mathcal{E}(L_1, \ldots, L_n)$ or (2) $\texttt{unsuc}(r, a) \leftarrow \mathcal{E}(L_1, \ldots, L_n)$, where r denotes a resource, a denotes the requesting agent, $L_1, \ldots, L_n$ are *trust modality literals* (i.e., built from `trust`, `distrust`, or `doubt`) or *cert literals* (i.e., built from `cert`), and $\mathcal{E}$ designates the syntactic structure of the employed connectives and quantifiers, where some suitable restrictions might apply.

We show a simple example for security policy rules and thereby give an informal semantics for the security policy specification language. Roughly speaking, we evaluate the truth values of trust modality literals with respect to the owner's local classification sets, and the truth values of cert literals with respect to the owner's local certificate set. Note that we here do not consider conflict resolution.

$$\texttt{suc(file2,}a_2\texttt{)} \leftarrow \exists a_1\ (\texttt{cert(}a_1\texttt{,Diploma,}a_2\texttt{)} \wedge \neg\texttt{distrust(}a_1\texttt{,Diploma))}$$

This rule derives a successful property conversion for a requesting participant a_2 concerning the bound property accessing `file2`, iff

- there exists a valid attribute certificate in $\mathcal{C}_o^f$ that binds the free property `Diploma` to the requesting participant a_2, whereby
- the issuer a_1 of this attribute certificate and the free property `Diploma` *must not* be a distrusted grantor-property pair, i.e., the grantor-property pair *must not* be member of the owner's classification set $\mathcal{K}_o^{DISTRUST}$.

6 Conclusion

Requesting resources in identity-less and distributed access control systems with autonomously acting agents requires that requesting agents present (sets of) certified properties in order to gain authorised accesses to the resources. Controlling agents base their access decisions not only on the presented certified properties, but rather on the correlation between certified properties and own expectations regarding both issuing and requesting agents. To address this problem, we have proposed a simple trust- and property-based access control model. The contribution of our paper lies in the identified correlations of certified properties, experiences and expectations of controlling agents, and resulting trust modalities with respect to decisions on property conversion and (direct) access requests. The work in this paper is still in progress. Examples of open issues include the examination of the proposed credentials/certificates types regarding their actual implementation. Further, our ongoing work explores elaborating the security policy specification language.

References

[1] Barker, S. and P. J. Stuckey, *Flexible access control policy specification with constraint logic programming.*, ACM Trans. on Information and System Security **6** (2003), pp. 501–546.

[2] Bertino, E., B. Catania, E. Ferrari and P. Perlasca, *A logical framework for reasoning about access control models.*, ACM Trans. on Information and System Security **6** (2003), pp. 71–127.

[3] Biskup, J. and Y. Karabulut, *A hybrid PKI model: application to secure mediation*, in: *Research Directions in Data and application Security, 16th Annual IFIP WG 11.3 Working Conference on Data and Application Security* (2003), pp. 271 – 282.

[4] Bonatti, P. A. and P. Samarati, *A uniform framework for regulating service access and information release on the web.*, Journal of Computer Security **10** (2002), pp. 241–272.

[5] Cantor, S., J. Kemp, R. Philpott and E. Maler, *Assertions and protocol for the oasis security assertion markup language (SAML), version 2.0*, http://docs.oasis-open.org/security/saml/v2.0/saml-core-2.0-os.pdf (2005).

[6] Carbone, M., M. Nielsen and V. Sassone, *A formal model for trust in dynamic networks*, in: *Proc. of the First International Conference on Software Engineering and Formal Methods* (2003), pp. 54 – 63.

[7] Ellison, C. M., B. Frantz, B. Lampson, R. Rivest, B. Thomas and T. Ylonen, *RFC 2693: SPKI certificate theory* (1999), ftp://ftp.isi.edu/in-notes/rfc2693.txt.
URL http://world.std.com/~cme/html/spki.html

[8] Grandison, T. and M. Sloman, *A survey of trust in Internet applications*, IEEE Communications Surveys & Tutorials **3** (2000).

[9] IETF, *Public Key Infrastructure (X.509)* (1998), iETF X.509 Working Group, www.ietf.org/html.charters/pkix-charter.html.
URL http://www.ietf.org/html.charters/pkix-charter.html

[10] Jajodia, S., P. Samarati, M. L. Sapino and S. S. V. *Flexible support for multiple access control policies.*, ACM Trans. on Information and System Security **26** (2001), pp. 214–260.

[11] Jøsang, A., R. Ismail and C. Boyd, *A survey of trust and reputation systems for online service provision*, Decision Support Systems (2007).

[12] Lee, A. J., M. Winslett, J. Basney and V. Welch, *Traust: A trust negotiation-based authorization service for open systems.*, in: *Proc. of the 11th ACM Symposium on Access Control Models and Technologies (SACMAT'06)* (2006), pp. 39–48.

[13] Li, N., W. H. Winsborough and J. C. Mitchell, *Design of a role-based trust-management framework*, in: *IEEE Symposium on Security and Privacy*, Berkley, California, 2002, pp. 114–130.

[14] OASIS, *eXtensible Access Control Markup Language (XACML), XACML 2.0 specification set*, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml (2007).

[15] Sprick, B. and S. Wortmann, *Time Dependent Trust Structures*, Computer Systems Science and Engineering, Special issue on TrustBus **20** (2005), pp. 411–419.

[16] Wang, L., D. Wijesekera and S. Jajodia, *A logic-based framework for attribute based access control.*, in: *Proc. of the 2004 ACM Workshop on Formal Methods in Security Engineering (FMSE 2004)*, 2004, pp. 45–55.