Dynamically Changing Trust Structure in Capability Based Access Control Systems

Sandra Wortmann, Barbara Sprick, and Christoph Kobusch

Information Systems and Security, Department of Computer Science,
University of Dortmund, Germany
{sandra.wortmann,barbara.sprick,christoph.kobusch}@udo.edu

Abstract. The functioning of modern IT-systems with autonomously acting components requires an elaborate access control system in which each participant can maintain her own trust structure.
In this work, we discuss ideas for an extension of capability based access control systems that allow the specification of dynamically changing trust of participants. We propose a classification of credentials and distinguish between credentials that have a positive and those that have a negative impact on access decisions. Furthermore, we investigate, how our ideas can be implemented in existing approaches for capability based access control systems.

1 Introduction

The functioning of distributed IT-systems requires an elaborate access control system. In a distributed IT-system with autonomous components, a fixed global or even hierarchical trust stucture is not suitable. In such a system, every participant maintains her own trust structure autonomously. Capability based access control systems are well suited to capture the individual and dynamically changing trust structure of each participant.

In a capability based access control system, access to a resource is granted or denied on the basis of the requester's capabilities rather than on the basis of her identity. Existing approaches for capability based access control systems such as [1–3] are monotonic: more certified properties usually imply more access permissions.

However, we believe that such a monotonic approach is too simple to reflect a substantial set of real world applications. Owners of resources might for example wish to explicitly prohibit other participants from accessing their resource or might wish to formulate exceptions from their general access control policy. In such cases, issued certificates can have a negative impact on the access decisions of owners of resources. In particular, we believe that not only certified properties and access permissions need to be considered: Each participant of a capability based access control system needs to dynamically maintain her own trust structure (concerning the trustworthiness of other participants). Consequently, revocation of already certified properties, explicit access prohibitions

S. Katsikas, J. Lopez, and G. Pernul (Eds.): TrustBus 2004, LNCS 3184, pp. 50–59, 2004.

and statements about trust and distrust concerning other participants need to be considered as well.

For these reasons we claim for an extended property based access control framework that is able to deal with a dynamically changing trust structure and with the potentially negative impact of certified properties. In our paper, we identify requirements for such a framework and suggest implemetation mechanisms.

Our paper is structured as follows. Section 2 describes the main roles in which participants of a credential based system may act and discusses their different interests and actions on the basis of an application scenario. Section 3 discusses aspects of time dependent, dynamically changing trust structures of participants. Section 4 analyses, how current approaches handle certificates with a potentially negative impact on access decisions. If certificates can have a negative impact on access decisions, one important question is how to enforce that all appropriate certificates are shown by a requester? Section 4 suggests implementation mechanisms as solution to this question. A discussion about related literature can be found in section 5. Finally, a conclusion is drawn in chapter6.

2 Capability Based Access Control

In capability based access control systems, access to resources is granted or denied on the basis of proven capabilities of the requester. Controllers of resources define the security policy of the resource in terms of capabilities or properties, participants of the access control system can certify properties to other participants who in turn can use the certified properties and capabilities to prove their eligibility for accessing a resource.

In the following we briefly introduce an application scenario and discuss the various roles in which participants of a capability based access control system can act and various types of certificates that can be issued.

2.1 Application Scenario

Consider a conference with two attached workshops. The conference as well as each of the workshops have their own online registration service. To register for the conference, one needs to prove membership of a university. To register for a workshop one has to be registered for the conference as well. Furthermore, it is only possible to register for one workshop. Consequently, if a person wants to register for a workshop, she needs to prove that she has registered for the conference but has not yet registered for the other workshop. People that verifiably have violated the guidelines for good research are excluded from participation in the conference and in the workshops. After successful registration, each of the registration services returns a registration receipt. Such a registration receipt can be used to request access to the respective conference or workshop site. Furthermore, a conference registration receipt is needed when registering for a workshop.

2.2 Roles and Their Interests

We distinguish among four different roles which the participants of the system can hold, namely *controller*, *assigner*, *grantee* and *verifier*.

Controllers are either owners of resources or their delegates. The main interest of a controller is to restrict access to the respective resource only to authorized participants. To do so, the controller defines the security policy of the resource in terms of capabilities and properties that authorize requesters for accessing the resource. Further, the controller certifies capabilities concerning access to this particular resource to other participants of the system. In our example, the organizing chairs of the conference or workshops acts the role of the controller when defining the access control policy of the registration web sites or issuing conference or workshop registration receipts.

Assigners act independently of particular resources. They autonomously certify properties to participants of the access control system. Usually, assigners do not have particular interests concerning the use of issued certificates. In our example, universities act as assigners when they certify university membership. These certificates are not bound to any particular purpose by the university.

Grantees collect certificates about their properties issued by the controller and assigners. Their main interest is to gain access to resources. When requesting access, grantees either present required certificates about their attributes or directly present authorization certificates. In our example, university members act as grantees when they collect certificates about their university membership or about conference and workshop registration.

Verifier grant or deny access to the particular resource on the basis of the resource's security policy and the requesters' certificates. In our example, the conference and workshop organizers act as verifiers. However, they have delegated the role of the verifier to the conference and workshop registration tools, respectively.

2.3 Certified Properties

As described in the previous paragraph, controllers and assigners issue certificates about certain properties to grantees. According to [4], we can distinguish between two different types of properties:

1. *Free properties* are certified by assigners. Their certification is not bound to any particular purpose and they do not directly entail an access permission at a particular resource.
2. *Bound properties* are certified by controllers of resources. Their certification is to be seen in the context of the respective resource. They express a promise about some specific access permission.

Note, that the distinction between free and bound properties is context dependent: In our example scenario, a registration receipt issued by the conference registration service certifies a bound property in the context of the conference web site. However, it can also be considered as a free property in the context of the workshop registration services.

3 Dynamically Maintained Trust Structure

As described in the previous section, in a capability based access control system, access to resources is granted to requesters on the basis of certified properties. Usually, an increase of certificates issued to a grantee implies an increase of access permissions, i.e. capability based access control systems are monotonic. However, we believe that a monotonic access control system is too restrictive to enable the specification of security policies of a substantial subset of real world applications.

The controller of a resource defines the resource's security policy, which in turn reflects the controller's trust structure of the access control system. In a monotonic capability based access control system, the controller defines the properties required for accessing a resource and constitutes which assigners are trusted to certify the required properties. However, the controllers also need a possibility to explicitly exclude holders of certain properties from access and to define a trust structure concerning assigners.

Apart from credential revocation mechanisms, there hardly exist any mechanisms in credential based access control systems that facilitate above mentioned non monotonic aspects of the controller's trust structure.

In many real world applications, the controller's trust structure is more complex and should reflect modalities such as *trust, distrust, belief* and *doubt* (concerning other participants and concerning certified properties). Often, the trust structure is not static but changes over the time. It is therefor desirable to have a time dependent notion of a trust structure that can be dynamically maintained by the respective controllers.

By certifying a free property, the assigner expresses her firm *belief* that the certified property holds for the grantee. As in monotonic capability based access control systems, a controller *trusts* certain assigners to certify certain free properties. This trust is reflected in the access control policy defined by the controller. If the controller certifies a bound property to a grantee, she expresses her *trust* in the grantee to appropriately use the access permission.

In some cases, a controller has reservations or *doubts* against participant for whom certain properties hold or against assigners of certain properties. These doubts should be expressed in the access control policies of the respective resources controlled by the controller.

If a controller explicitly *distrusts* particular participants, access for these participants should be explicitly prohibited.

In section 2.3, we distinguished between free and bound properties. Speaking in terms of SPKI/SDSI, certificates about free properties are called attribute credentials and certificates about bound properties are called authorization credentials. Attribute credentials refer to belief of assigners concerning properties of grantees, authorization credentials refer to trust of controllers concerning eligible and appropriate use of resources.

To be able to additionally express doubt and distrust in a credential based way, we suggest to consider another type of credentials, namely *prohibition credentials*.

As we have argued before, the trust structure of controllers may change over the time, an appropriate credential based access control framework should allow to dynamically maintain trust structures and in particular, to allow to revoke previously issued credentials. We thus suggest to consider a fourth type of credentials called *revocation credentials*.

3.1 Types of Credentials

Attribute Credentials
By issuing an *attribute credential*, an assigner certifies that the grantee holds the specified free property. If the assigner wants to certify the absence of a particular free property p, she issues an attribute credential certifying, that the grantee has property `not` p. This type of attribute credential, whether it certifies a property p or the absence of property p is not bound to a particular resource. It can be used for requesting access at any arbitrary resource, depending on the resources security policy. Note, that a certificate about the absence of a particular property does not necessarily have a negative impact on the access decision. It might well be, that exactly the absence of the property is required for access. Recall the example about the conference management scenario. Only users who can prove that they have *not* registered for workshop a are entitled to register for workshop b.

Authorization Credentials and Prohibition Credentials
By issuing an *authorization credential*, a controller explicitly certifies that the grantee is eligible to access the resource. For instance, the users holding a conference registration receipt are entitled to use the conference web site. By issuing a *prohibition credential*, a controller explicitly certifies that the grantee is prohibited from accessing the resource. A prohibition credential has a negative impact for the grantee on the access decision to the resource. If, for example, a user has verifiably violated the guidelines of good research, the organization chair explicitly excludes the user from registration.

Revocation Credentials
As motivated before, assigners and controllers may want to revoke previously issued credentials as their trust structure may change over the time.

We can distinguish between two cases of changing belief: In the first case, the issuer of a credential knows at issuing time, that the certified property is valid only until a particular point in time or at least, that she wants to certify the association between the grantee and the property only for a particular time period. In this case, she can simply certify this by issuing a credential which is valid only for this particular period in time. For example, a university issues student certificates only for one semester and membership certificates for scientific staff only for the time of their contract. In the second case, the issuer of a credential learns only after certifying a property, that the grantee of the credential does no longer hold the certified property. In this case, the issuer of the credential will want to revoke the issued certificate. For this purpose, a *revocation credential* can be issued stating that the formerly issued credential is no

longer valid. For example, when learning, that a user who has already registered for the conference has violated the guidelines for good research, the conference chair may want to revoke the previously issued registration receipt. Note, that not only authorisation credentials and attribute credentials can be revoked but also prohibition credentials. Thus, a revocation credential can have both positive and negative impacts on access decisions.

Negatively Used Attribute Credentials
The access control policy of a resource should not only define necessary access conditions but also conditions that exclude from access. It is desirable to be able to define a policy that allows access for all requesters having property a except for those having property b. By issuing attribute credentials, an assigner subsumes groups of grantees that have the same property. In order to exclude a subset of such a group from access, the controller can again identify the subset that is to be excluded by a set of attribute credentials. In such a case, an attribute credential can have a negative impact for the grantee on the access decision. The controller subtracts the group of grantees determined by the attribute credentials from the group of grantees eligible for access to a resource. If a grantee holds a workshop registration certificate for workshop b, this attribute credential has a negative impact on the access decision for the registration service of workshop a.

4 Implementation of Doubt and Distrust

This section surveys how doubt and distrust can be implemented in current public key infrastructures ([3, 1, 5, 2]).

Revocation Credentials
Mechanisms Suppose, an issuer wants to revoke a previously issued credential as she does no longer belief, that the grantee of the credential holds the certified property. The KeyNote Trust Management System does not currently provide credential revocation mechanisms. However, an issuer of a KeyNote credential may specify and implement revocation policies. In other public key infrastructures, e.g. X.509 or SPKI/SDSI, the issuer of a credential may give further validity conditions. The revocation of credentials is usually specified in certificate revocation lists (CRL). Such lists need to be checked by the verifier of a resource for access decisions: revoked credentials should not have any impact on the access decision.

Implementation CRLs are usually placed on designated servers. Because of the potential length of such lists it is sometimes more appropriate to issue signed *δ-CRLs* that contain only the difference between the current CRL and the previously issued CRL. The Online Certificate Status Protocol (OCSP), [6], improves standard CRLs by avoiding the transmission of long CRLs and by providing more recent revocation information. To do so, it uses so-called status requests for credentials. In [7], Kocher suggested Certificate Revocation Trees. Such a data structure is a hash tree where the leaves denote the currently revoked credentials.

Prohibition Credentials
A prohibition credential explicitly prohibits the holder of the credential from accessing the respective resource. Existing public key infrastructures do not currently provide mechanisms to implement prohibition credentials. One of the main questions to answer is why would a user present a prohibition credential to the verifier?

Negatively Used Attribute Credentials
Mechanisms The access control policy of a resource specifies access requirements on the basis of attribute credentials. Note, that attribute credentials can have both positive and negative impacts on the access decision: While some attributes are mandatory, others may not desirable and thus exclude form access. Again, the question arises why users would present credentials that have a negative impact on the resource's access decision? Existing public key infrastructures do not provide appropriate mechanisms for enforcing such "negatively used" credentials.

Implementation. On the specification side, we suggest, that the controller defines the security policy through algebra expressions built from free properties and operators. To specify negatively used credentials, the controller may use a subtraction operator. Roughly speaking, the semantics of such an algebra expression would be to interpret attribute credentials, certifying free properties, as groups of grantees having the respective properties. The operators are then evaluated as set-theoretical operations applied to sets of grantees. Negatively used attribute credentials are standard attribute credentials, but are negatively interpreted when used as subtrahend in the underlying security policy.

As mentioned before, on of the main problems is how to enforce grantees to show attribute credentials when thy have a negative impact on access decisions. Biskup and Wortmann ([8]) propose a solution to this problem: The authors suggest a new kind of online test of a credential as so-called *location* that is used in combination with a new kind of subject of a credential as so-called *first-of-two*. An alternative approach to prevent grantees from hiding negatively used attribute credentials, investigated in [9], introduces so-called *not credentials* that certify a grantee her "not membership" of a particular group.

5 Related Work

We have focused on credentials that certify participants non identifying capabilities. While SPKI/SDSI and KeyNote are based on public keys of the participants and allow for a non identifying approach, the X.509 public key infrastructure [1] does not fully support this non identity based approach as credentials are inevitably identifying in X.509.

We analyzed attribute credentials, authorization credentials, prohibition credentials and revocation credentials. The differentiation between attribute credentials (certifying free properties) and authorization credentials (certifying bound properties) leads us to the public key infrastructure SPKI/SDSI, because the KeyNote trust management system [3] does not support attribute credentials.

SPKI/SDSI was invented in 1996 and results from a name definition part called Simple Distributed Security Infrastructure (SDSI [5]) and an authorization part called Simple Public Key infrastructure (SPKI [2]). A lot of work contributed to a semantics for SPKI/SDSI. Comparing Abadi's logic, introduced in [10], to the requirements and aspects identified in our paper, his modal operator *says* expresses the belief of an issuer (controller or assigner) about the properties of other participants. Translated to our setting, Abadi's relation $A \Rightarrow B$, read as "(participant) A speaks for (controller) B", expresses controller B's promise of an access right towards participant A, or controller B's trust towards participant A. Howell and Kotz [11] extend Abadi's logic by (restricted) delegation and authorization. In their extension, belief of an issuer (controller or assigner) is modeled by the modal operator `believes`. The formula A `believes` σ, where A is an issuer and σ is a certificate certifying a grantee to have a certain property, can be interpreted as "Issuer A believes the binding expressed in certificate σ to be true". Howell et.al. further introduce a relation $A \overset{T}{\Rightarrow} B$, read as "(participant) A speaks for (controller) B regarding (the set of access permissions) T". Interpreted in our setting, this formula expresses controller B's trust in (participant) A regarding the set of access permissions T. In [12], Halpern and van der Meyden develop a logic to deal with SPKI authorization credentials. However, their logic does not provide an mechanism for the specification of attribute credentials. Thus, it only supports the specification of certificates about bound properties, but not about free properties. logic programming based semantics for SPKI/SDSI and in [13] Li and Mitchell introduce a first order logic semantics of SPKI/SDSI. Most of the logics are able to express belief and (restricted) trust of participants of a capability based access control system. Some of the languages provide mechanisms for treating a dynamically changing trust structure, see e.g.[11]. However, none of the logics explicitly formalizes prohibition or revocation credentials.

Some work has been done about the meaning of credentials and their revocation, see for instance [14–17]. In particular [17] introduces a language for creating and manipulating, i.e. issuing and revoking, credentials. All approaches deal with revocation of credentials, some of them treat issues of time, e.g. [14]. However, as to our knowledge, there do not exist any approaches that deal with negatively used credentials in general or prohibition credentials in particular.

6 Conclusion

Capability based access control systems have shown to be appropriate for access control in highly distributed systems where a global controlling instance cannot be assumed. However, current implementations of capability based access control systems, such as [3], [2] or [1] are monotonous and have significant limitations when it comes to access prohibitions. In this paper, we first analyzed various roles in a capability based access control systems and discussed their interests. We pointed out the need for appropriate mechanisms for assigners of credentials and controllers of resources to dynamically and autonomously maintain their

trust structures. Further, we suggested new types of credentials that are suited to help assigners and controllers specifying and maintaining their trust structures. Finally, we discussed how credentials with a potentially negative impact on access decisions can be implemented in current credential based access control systems.

Acknowledgements

We thank Joachim Biskup and Torben Weibert for helpful discussions and proof reading. A part of this work was funded by the DFG German Research Council (DFG) under grant number BI 311/11-1.

References

1. IETF: public key infrastructure (x.509). http://www.ietf.org/html.charters/pkixcharter.html (1998) IETF X.509 Working Group.
2. Ellison, C., Frantz, B., Lampson, B., Rivest, R., Thomas, B., Ylonen, T.: SPKI certificate theory. Internet RFC 2693 (1999)
3. Blaze, M., Feigenbaum, J., Ioannindis, J., Kermytis, A.: The keynote trust management system version 2. http://www.cis.upenn.edu/~keynote/Papers/rfc2704.txt (1999) IETF RFC 2704.
4. Biskup, J., Karabulut, Y.: A hybrid pki model: Application to secure mediation. In: Research Directions in Data and Application Security. 16th Annual IFIP WG 11.3 Working Conf. on Data and Application Security, Kluwer, Boston etc. (2003) 271–282
5. Rivest, R., Lampson, B.: SDSI – a simple distributed security infrastructure. http://theory.lcs.mit.edu/~cis/sdsi.html (1996)
6. Myers, M., Ankney, R., Malpani, A., Galperin, S., Adams, C.: X.509 internet public key infrastructure online certificate status protocol – OSCP. http://www.ietf.org/rfc/rfc2560.txt (1999) IETF RFC 2560.
7. Kocher, P.: On certificate revocation and validation. In: Proceedings of the 2nd Internation Conference on Financial Cryptography (FC'98). Volume 1465 of LNCS., Springer Verlag, Berlin (1998) 172–177
8. Biskup, J., Wortmann, S.: Towards a credential-based implementation of compound access control policies. In: Proceedings of the 9th ACM Symposium on access control and models (SACMAT), Yorktown Heights, New York, USA, ACM (2003) To appear.
9. Kobusch, C.: Mechanismen zur durchsetzung negativ wirkender zertifikate in zugriffskontrollsystemen. Master's thesis, University of Dortmund (2003) http://ls6-www.cs.uni-dortmund.de/dpa.html.
10. Abadi, M.: On SDSI's linked local name spaces. In: Proceedings of the 10th IEEE Computer Security Foundations Workshop, IEEE Computer Society (1997) 98–108
11. Howell, J., Kotz, D.: A formal semantics for SPKI. In: Proceedings of ESORICS 2000. Volume 1895 of LNCS., Springer Verlag, Berlin (2000) 140–158
12. Halpern, J., van der Meyden, R.: A logic for SDSI's linked local name spaces. Journal of Computer Security **9 (1–2)** (2001) 47–74
13. Li, N., Mitchell, J.: Understanding spki/sdsi using first order logic. In: Proceedings of 16th IEEE CSFW, Boston, Mass., ACM Press, New York (2003) 182–189

14. Li, N., Feigenbaum, J.: Nonmonotonicity, user interfaces, and risk assessment in certificate revocation. In: Proc of the 5th Int. Conf. on Financial Cryptography (FC'01). Volume 2339 of LNCS., Springer Verlag, Berlin (2001) 166–177
15. Rivest, R.: Can we eliminate certificate revocation lists? In: Proceedings of the 2nd Internation Conference on Financial Cryptography (FC'98). Volume 1465 of LNCS., Springer Verlag, Berlin (1998) 178–183
16. Myers, M.: Revocation: options and challenges. In: Proceedings of the 2nd Internation Conference on Financial Cryptography (FC'98). Volume 1465 of LNCS., Springer Verlag, Berlin (1998) 165–171
17. Gunter, C., Jim, T.: Generalized certificate revocation. In: Proceedings of the 27th ACM SIGPLAN-SIGACT symposium on Principles of programming languages, ACM (2000) 316–329
18. Myers, M., Malpani, A., Pinkas, D.: X.509 internet public key infrastructure online certificate status protocol, version 2. http://www.ietf.org/proceedings/02mar/ID/draft-ietf-pkix-ocspv2-02.txt (2002) Network Working Group.