Semantic Access Control Model: A Formal Specification[*]

Mariemma I. Yagüe, María-del-Mar Gallardo, Antonio Maña

Dpto. de Lenguajes y Ciencias de la Computación
University of Málaga, 29071 Málaga, Spain

Abstract. The Semantic Access Control Model (SAC), built on the basis of separation of the authorization and access control management responsibilities, provides adequate solutions to the problems of access control in distributed and dynamic systems with heterogeneous security requirements. SAC is characterized by its flexibility for accommodating dissimilar security policies, but also by the ease of management and control over a large number of distributed elements and the support for interoperability of authorization mechanisms. In this paper, we present the semantic validation algorithms developed in SAC to detect semantically incomplete or incorrect access control policies. Additionally, the formal model of SAC along with some proofs of its soundness is introduced. This formalization is the basis for additional model checking of the semantic validation algorithms developed.

Keywords: Access Control, Authorization, Distributed Systems Security, Formal Methods in security.

1 Introduction

When security requirements for distributed applications are considered, *authorization* often emerges as a central element in the design of the whole security system. Many other security requirements depend on the flexibility, trustworthiness and expressiveness of the authorization scheme. On the other hand, *access control* is the mechanism that allows resource owners to define, manage and enforce the access conditions for each resource [16]. These two concepts are very closely related because authorizations are usually the basis for the access decision in access control systems.

The notions upon which an access control model is defined determine its flexibility to be applied in different environments and systems. Traditional access control models have been designed to provide access control in some specific scenarios. However, the mechanisms provided by these models are not expressive enough to deal with very dynamic environments with a high volume of heterogeneous data, where new resources are incorporated to the system continuously,

[*] Work partially supported by the Spanish Ministry of Science and Technology under the Research Projects PRIVILEGE (TIC2003-08184-C02-01) and SELF (TIN2004-7943-C04-01)

each resource possibly needing a different access control policy, and where policies may change frequently. Furthermore, traditional access control schemes are not suitable for scenarios where the local registration and authorization of users is not appropriate or with a very large number of heterogeneous registered users. In these systems, for scalability reasons, it is not practical to keep access and authorization information for each user.

In this paper, we present the formalization of a more general access control model developed for these new environments. The *Semantic Access Control* (SAC) model [20] was especially designed for handling the access control in heterogeneous, distributed and large environments. This model solves the above mentioned scalability problems, facilitates access control management, and provides a means to express access conditions in a natural and flexible way.

SAC considers the operation of several independent access control systems and authorization entities. The access control to resources is independent of their location. Additionally, the identification of the user or client is not mandatory. On one hand, the client possesses a set of attributes and, on the other hand, the access control to resources is based on the specification of a set of attributes that the client has to present to gain access to them. For interoperability and security reasons, client attributes must be digitally signed (in the form of an attribute certificate) by a trusted certification entity, external to the access control management system. The independence of the certification of attributes is the key to the interoperability achieved because it allows attributes to be safely communicated avoiding the necessity of being locally recorded by the system administrator. Additionally, this approach avoids the registration phase of the client, and the recording of a client attribute for each access control system. For this approach to be secure, a mechanism to establish the trust between these access control systems and the authorization entities was required. We addressed this problem using semantic information about the certifications issued by each authorization entity. One of the main characteristics of the SAC model is that, as opposed to traditional schemes, the attributes required to access a resource may depend on the semantic properties of the resources. The allocation of the policy corresponding to a resource is not based on the storage structure of resources but on their semantic properties. Of course, it is also possible to consider the structure of storage.

An orthogonal problem when defining an access control model is to assure that it is semantically sound. In this context, soundness means that users not fulfilling the access policy cannot access resources. SAC enables the semantic validation of the access control policies. Additionally, in order to prove soundness, we have formalized the SAC model using inference rules. The construction of the formal model makes use of the semantic information handled through the different SAC metadata models [23]. In this formalisation, we have defined two different entailment relations: the first one is able to infer whether a target satisfies an attribute; the second relation deals with the access policies. In this last case, we have introduced some useful operators that combine different access rules. It is worth noting that the formal model presented has inherited the flexibility of SAC in the sense that, if necessary, we may add new operators transparently.

In summary, SAC was developed to facilitate the management of complex access control systems, while guaranteeing the simplicity, correction and safety of the system. To deal with this, SAC provides a set of algorithms for the automatic validation of the access control policies defined by the system administrator. This work presents the formal basis to prove its correctness.

This paper is organized as follows. Section 1 presents some related works. Afterwards and before introducing the formal model of SAC, Section 2 shows some background on SAC which helps its formalisation. Section 3 introduces the formal model of SAC with the formalization of the Source of Authorizations of a PMI and the derivation rules to deduce information from certificate classes. We finalize with some conclusions and projected work for the near future. Lastly but not least, an example and proofs of theorems are illustrated in the appendices.

2 Related Works

Traditional access control models such as Discretionary (DAC) [1], Mandatory (MAC) [15] and Role-Based (RBAC) [17] Access Control were developed for closed environments. Consequently, they are built on the basis of modelling the environments that motivated their development [23]. Among these models, RBAC is commonly accepted as the most appropriate paradigm for the implementation of access control in complex scenarios. RBAC can be considered a mature and flexible technology. In RBAC, the structure of groups is defined by the security administrator and it is usually static. Although grouping users can suffice in many different situations, it is not flexible enough to cope with the requirements of more dynamic systems where the structure of groups can not be anticipated by the administrators of the access control system. In these scenarios, the structure of the system may be increased dynamically with new resources which may possibly need a different group structure and access control policy. Additionally, the policy for a given resource may change frequently.

We believe that a more general approach, such as the one presented by the Semantic Access Control model, is needed in order to properly deal with these new environments. For example, in the referred situations, groups are artificial substitutes of a more general tool: the attribute. In fact, groups are usually defined on the basis of the values of some specific attributes (employer, position, $\cdots$). Some attributes are even built into most of the current access control models. This is the case of the user element; the identity is just one of the most useful attributes, but it is not necessary in all scenarios and, therefore, it should not be a built-in component of a general model. Recent literature in the area of access control for distributed heterogeneous resources from multiple sources shows the use of attribute certificates and PMIs. Firstly, we highlight two research projects, Akenti [7] and Permis [5]. Akenti Project proposes an access control system to restrict access to distributed resources controlled by multiple stakeholders. The requirement for the stakeholders to trust the rest of the servers in the network, as well as some security vulnerabilities related to the existence of positive and negative use-conditions, are the main drawbacks of Akenti. The PERMIS Project [13] objective is to set up an integrated infrastructure to solve identification and authorization problems. A specific goal is to specify the authorization policy in a language that can be both easily parsed by computers and read by the security administrators with or without software tools. The PERMIS group concluded that XML is the most appropriate candidate for a policy specification language. However, because PERMIS system is based on the RBAC model, it shares its limitations. Moreover, the requirement of supporting a PKI is hard to fulfil and it is not necessary in many authorization scenarios.

Regarding the different XML-based languages proposed for access control, digital rights management, authentication and authorization, many similarities and interesting features can be found among them. Some other features, such as policy parameterisation and composition are not supported. Moreover, some features provided by those languages are not appropriate in heterogeneous and dynamic scenarios. Two relevant proposals for access control to XML documents are the Author-X system [2] and the FASTER project [6]. They differ from SAC in that both systems have been specifically developed for XML documents, unlike the general definition of resource in this work. Author-X is based on credentials that are issued by the access control administrator. Therefore, in practice, each credential will be useful only for a single source, limiting interoperability. A direct consequence of this approach is that users must subscribe to sources before they can access their contents. In the Semantic Access Control Model (SAC) however we have semantically integrated a Privilege Management Infrastructure that will be responsible for issuing digitally signed attribute certificates. Another relevant proposal is XACML [14], an OASIS standard that proposes two XML-based languages to describe access control policies and access decision requests and responses. Although XACML and SAC share some similarities, there are important differences [21].

Other access control languages have been developed in the security community to support different access control approaches. Jajodia et al. present in [9] a logical language which allows users to specify the policy according to what access control decisions are to be made as well as the authorizations. SAC is focused in this direction, but in the SAC case we are interested in access control for highly dynamic systems with an important volume of heterogeneous data and multiple independent data sources. We use XML and XML Schema to enable the definition of policies expressed by means of rules and the representation of derivation rules for the attribute classes used in the policies semantic validation.

Some works have used formal semantics for policy representation and evaluation such as [19] but this work differs from ours in that they address issues such as positive and negative authorizations. Another interesting work is the Policy Maker system [3, 4], which focuses on construction of a practical algorithm for determining trust decisions. The main drawback of this proposal is the use of a policy language with a low abstraction level and it is very cumbersome, unlike the SPL policy language defined in SAC.

Finally, we must highlight an innovative feature presented by SAC which is semantic and contextual validation of policies. In SAC we have taken into account that the creation and maintenance of access control policies is a difficult and error prone activity. Therefore, in the design of SAC we have considered that this access control model must facilitate and guarantee the correct administration of the system. To reach this objective, a set of algorithms have been defined to detect incorrect access control policies. The semantic algorithms carry out inference processes using the rules defined in the Source Of Authorization Description (SOAD) documents and have been implemented as part of the Semantic Policy Validator (SPV) tool.

3 Fundamentals of the Semantic Access Control Model (SAC)

Most of current access control schemes base their authorization approaches on locally-issued credentials that are based on user identities. This type of credential presents many drawbacks. Among them we highlight:

(a) they are not interoperable;
(b) the same credentials are issued many times for each user, which introduces management and inconsistency problems;
(c) credentials are issued by the site administrator, however, in most cases, the administrator does not have enough information or resources to establish trustworthy credentials; and
(d) they depend on user identity. However, in practice, frequently the identity of the user is not relevant for the access decision. Sometimes, it is even desirable that the identity is not considered or revealed. Furthermore, in systems based on identity, the lack of a global authentication infrastructure (a global Public Key Infrastructure, PKI) forces the use of local authentication schemes. In these cases, subscription is required and users have to authenticate themselves to every accessed source.

To solve the aforementioned problems, single-sign-on mechanisms are becoming popular [18]. Although these mechanisms represent an improvement, they do not enable interoperability while maintaining the diversity. The reason is they are based on federation of sources and all federated sources must agree on a homogeneous access control scheme. Additionally, credentials remain local, not to a site, but to a set of them.

On the other hand, digital certificates [8] can securely convey authorizations or credentials. Attribute certificates bind attributes to keys and make authorizations interoperable and mobile, since attribute certificates can securely transport authorization information. This mobility provides the foundation for a better alternative to actual Single Sign-On schemes.

Another important advantage of attribute certificates is that they can be used for various purposes. They may contain group membership, role, clearance, or any other form of authorization. As a consequence, digital certificates provide means for the deployment of scalable and flexible access control schemes, since access conditions are expressed in terms of sets of attributes instead of users or groups. Users must possess attribute certificates attesting that they meet the requirements. As opposed to traditional access control schemes, a high number of users and attributes do not degrade performance and manageability of this solution.

On the other hand, when discussing how to establish the access conditions applicable to a particular resource, two main approaches must be considered: (i) conditions are established on the basis of the location of the resources or, (ii) conditions are based on the properties of the resources. The fact is that conditions and restrictions of access naturally depend on the semantic properties of the target resource that are neglected in structure-based approaches. Therefore, an approach based on semantic descriptions of the contents is much more flexible and natural. Moreover, it is easy to incorporate structure-based requirements in the semantic model. Additionally, the structure is much more volatile than the semantics. The incompatibility between the structure required for the application domain and the ones that match the security requirements confirms that structure-based approaches are not able to represent these situations in a natural way.

Another drawback of structure-based approaches is that the number of policies becomes very large. In fact, these approaches usually imply the definition of several policies for each resource. Positive and negative authorizations are used in these cases to facilitate the definition of simple policies and to reduce the number of policies. The price to pay is the presence of ambiguities, which in turn requires the definition of conflict resolution rules. Consequently, the administration of the system becomes complex and difficult to understand, increasing the chance of incorrect policies being produced.

The Semantic Access Control model (SAC) [20] was developed following a different approach. It was called this because semantics are the basis of the access conditions and its design follows a semantic approach. The SAC model is based on the semantic properties of the resources to be controlled, properties of the clients that request access to them, semantics about the context and finally, semantics about the attribute certificates trusted by the access control system. The semantic-based and modular approach adopted in SAC, facilitates the definition and management of policies avoiding the use of positive and negative authorizations. Tools provided to support the policy specification, composition and validation also serve this objective. The Semantic Access Control model has been implemented on the basis of the Semantic Policy Language (SPL) to specify the access control criteria and the semantic integration of an external authorization entity.

3.1 Semantic Policy Language (SPL)

SPL XML-Schema based policy definition language [24] was designed to specify policies in a simple way, enabling high level expressiveness and efficient evaluation. Usual components of access policies include the target resource, the conditions under which access is granted/denied and, sometimes, access restrictions. As opposed to other languages, specifications in SPL do not include references to the target object. Instead, a separate specification called *Policy Applicability Specification* (PAS) is used to relate policies to objects dynamically when a request is received. Both SPL Policies and PAS use semantic information about resources, included in *Secured Resource Representations* (SRRs), and other contextual information documents.

SPL Policies and PAS can be parameterised allowing the definition of flexible and general policies, thus reducing the number of different policies to be managed. Parameters, which can refer to complex XML elements, are instantiated dynamically from semantic and contextual information. Additionally, policies can be composed, importing components from other policies without ambiguity. This compositional approach allows us to define the abstract meaning of the elements of the policies, providing a mechanism to achieve abstraction, which also helps in reducing the complexity of management.

The schema for SPL specifications is represented as a set of XML-Schema templates that facilitate the creation of these specifications, allowing their automatic syntactic validation [24]. SPL policies can include components defined locally as well as imported elements. The ability to import elements enables the modular composition of policies based on the XPath standard. An SPL Policy is composed of a set of *access_Rule* elements. Every *access_Rule* defines a particular combination of attribute certificates required to gain access, associated with an optional set of actions (such as *Notify_To*, *Payment* and *Online_Permission*) to be performed before access is granted. In this way, provisional authorization or PBAC [10] is enabled in SPL.

3.2 Semantic Description of the Sources of Authorization (SOAD)

As we have already mentioned, one of the basis of SAC is the separation of the certification of attributes and access control management responsibilities, in order to build a scalable and flexible solution.

A *Privilege Management Infrastructure* (PMI) [8] provides attribute certification services. It is then reasonable to expect that the PMI includes different certification authorities (SOAs), each one with a well-defined certification domain. That is, each SOA should be authoritative for a limited set of attributes and users. Ideally, each attribute would be certified only by one SOA. This raises the issue of the interoperability of the attribute certificates.

For example, suppose that `Peter Smith` is an authorized broker at the Chicago Board of Trade. Then `Peter` will have two separate certificates: an identity certificate attesting to his identity information and an attribute certificate attesting to his being an authorized broker at the Chicago Board of Trade. Both certificates can be related, for instance, by including the serial number and/or a hash value of the identity certificate in the attribute certificate. Suppose now that our friend `Peter Smith` is also member of the `Chicago Siesta Club` (`CSC`), a `public library`, `Greenpeace`, etc. If centralized access control schemes are used in these institutions, each one will have to locally register the different attributes of Peter Smith that are applicable to their access control policies. For instance, if the CSC has a discount for Greenpeace members then it is necessary to record Peter 's Greenpeace membership in the local database of users of CSC. However, how can CSC be sure that Peter is member of Greenpeace? What if Peter leaves Greenpeace? How does CSC know about this?

On the contrary, if the attribute certification function is separated then access control systems responsibilities are limited to establishing the local access control policies, making the system simpler, more dynamic and flexible, and more secure. Obviously, this approach requires that the access control system is complemented by an external component providing certification functions. The PMI is precisely that component. A consequence of the separation of access control and authorization functions (now provided by the PMI) is that the access control administrators do not have control over some factors that are used in their access control systems. Consequently, a mechanism to establish the trust between these administrators and the PMI is required.

In SAC, we addressed this problem using semantic information about the certifications issued by each SOA. This assists the security administrators in the creation and semantic validation of access control policies. In SAC, every SOA produces and digitally signs a set of Source Of Authorization Descriptions (SOADs) that express the semantics of the attribute certificates it issues [22]. These metadata documents describe the different attributes certified by a SOA, including names, descriptions and relations of attributes. SOADs are used to establish the trust between the PMI and the access control systems. They convey the information needed by the access control system to understand the semantics of the attribute certificates, which is essential in order to take appropriate access decisions.

3.3 Semantic Validation of Policies

The information contained in SOADs is also essential for the semantic validation of the policies, enabling the detection of semantically incomplete (or incorrect) policies through a Semantic Policy Validator (SPV) tool developed with this objective [20]. The SPV makes inference processes using the rules defined in the SOAD documents. The semantic validation ensures that the policies written by the security administrator produce the desired effects. An interesting feature of the SPV is that it allows policies to be validated in the context where they will be applied. The use of semantic information about the context allows the administrator to include relevant contextual considerations in a transparent manner. The SPV can perform three types of validations:

1. Test Case Validation: Given a request to access a resource and a set of attribute certificates, this algorithm outputs the sets of attribute certificates needed for accessing that resource. Most of the time, this feature will be used to check that a set of attribute certificates is incompatible with the access criteria for that resource. For instance, the administrator of our university

can use this validation to guarantee that it is not possible for a student to access a given resource (i.e., documents containing marks). During the validation process, the SPV generates the sets of attribute certificates that are not excluded by the input set, and checks the generated ones against all possible combinations of attribute certificates that grant access to the resource.

2. Access Validation: Given a request to access a resource, this algorithm outputs the sets of certificates that grant access to that resource. For this validation process, the SPV generates the policy for the resource and all sets of attribute certificates equivalent to those required by the policy.
3. Full Validation: The goal of this process is to check which resources can be accessed given a set of attribute certificates. Therefore, SPV generates the policy for each resource and, afterwards, all attribute certificates that can be derived from the input set of attribute certificates. Finally, it informs of every resource that can be accessed using the input attribute certificate set.

4 Formal Model of the Semantic Access Control

In this section, we formalize the deductive approach followed by the SAC model in order to grant/deny a request to access a given resource.

A *target* is any entity that may hold properties. In the SAC model, targets may be clients or resources. Properties of the targets are called *attributes*. Let T and $\mathcal{A}$ be, respectively, the sets of all possible targets and (atomic) attributes in a given application domain. We assume that each attribute $a \in \mathcal{A}$ has a negative counterpart $\neg a \in \mathcal{A}$ denoting the opposite attribute. For instance, attribute "non-student" is the negative counterpart of "student". In addition, we suppose that $\neg\neg a = a$. The first step to formalize SAC is to associate each target t with the set of attributes it holds at every instant in time. To this end, we define the set $\mathcal{A}^* = \mathcal{A} \cup \{!a | a \in \mathcal{A}\}$.

Function $K : \mathbb{N} \to (T \to \wp(\mathcal{A}^*))$[1] defines the true attributes held by targets in each time instant as follows:

- $a \in K(m)(t)$ means that target t holds attribute a at time instant m.
- Targets cannot hold simultaneously an attribute a and its negation $\neg a$ in an specific time instant m. Thus, $a \in K(m)(t) \Rightarrow \neg a \notin K(m)(t)$. On the other hand, it is possible that some attributes cannot be associated to certain targets. Thus, it may be that both $a \notin K(m)(t)$ and $\neg a \notin K(m)(t)$ hold. For instance, it makes no sense to apply attributes "divorced"/"non-divorced" to a "printer".
- Operator ! is a weak version of $\neg$ whose meaning is given as

$$!a \in K(m)(t) \text{ \textit{iff} } a \notin K(m)(t) \tag{4.1}$$

That is, $!a \in K(m)(t)$ means that t does not hold attribute a at time instant m. But it says nothing about $\neg a$. However, when $\neg a \in K(m)(t)$, following

[1] $\wp(\mathcal{A}^*)$ denotes the powerset of set $\mathcal{A}^*$.

the previous discussion, we have that

$$\neg a \in K(m)(t) \Rightarrow !a \in K(m)(t) \tag{4.2}$$

It is worth noting that time is introduced in the formal model because attributes held by targets may vary with time. Thus, it is possible for a target to hold an attribute a in a given instant m and to hold $\neg a$ in some future instant $f > m$. For example, target `María` may currently have the attribute "student", but it is very probable that, in the future, when she finishes her studies, `María` holds attribute "non-student". In order to properly deal with time, we assume that function $ctime :\to \mathbb{N}$ returns the current time instant.

In contrast to the *true facts* represented by function K, the SAC model makes use of SOAs to certify such facts. In other words, SOAs are the formal artefact devoted to providing certificates about targets that must be consistent with the reality represented by function K. In the rest of this section, we formally define how SOAs infer information about targets when required.

4.1 Formalizing SOAs

As mentioned above, a Source of Authorization k.a. SOA is a certification entity responsible for issuing *attribute certificates* attesting to a set of properties about targets. Each SOA has a *certification domain*, i.e. a set of targets and properties that can be certified by this SOA. For instance, the SOA of a university may issue certificates related to the enrollment of its students in courses, but not about their marital status. Likewise, it can not issue certificates related to the enrollment of students from other universities. Let $\mathcal{S}$ be the set of all SOAs in a given domain. In the sequel, we will use symbols σ, τ, etc. as elements of $\mathcal{S}$.

Given a SOA $\sigma \in \mathcal{S}$, an attribute certificate signed by σ is an expression of the form $\sigma\langle\langle a, t\rangle\rangle_d$, where d represents the temporal limit of the validity of the sentence. Thus, $\sigma\langle\langle a, t\rangle\rangle_d$ means that σ certifies that target t holds attribute a from the current time instant until the validity of the certificate expires in time d. We assume, without loss of generality, that the holder t of this attribute certificate will be identified by its public key[2]. Let $T_\sigma \subseteq T$ be the set of all targets in the certification domain of σ.

Besides attribute certificates, in our model, SOAs also provide rules (represented in SOAD metadata documents) defining semantic relations among attributes using the so-called *certificate classes*. Given an attribute $a \in \mathcal{A}^*$, the certificate class $\sigma\langle\langle a\rangle\rangle$ is used by the rules to express that SOA σ is responsible for checking attribute a. Thus, the notation of *certificate classes* allows us to easily represent the *trust relationship* among SOAs.

Formally, each SOA $\sigma \in \mathcal{S}$ is constituted by a 3-uple $\langle \mathcal{D}_\sigma, \Sigma_\sigma, SOAD_\sigma \rangle$ where

[2] In asymmetric encryption schemes each user has a pair of related keys. One of these keys, the Public Key, is publicly distributed while the other one, the Private Key, must be kept secret. Public Key's are included in digital certificates, so that other users can verify their authenticity.

1. $\mathcal{D}_\sigma \subseteq \mathcal{S}$ is the set of SOAs in which σ trust to delegate the task of issuing attribute certificates. We assume that $\sigma \in \mathcal{D}_\sigma$.
2. $\Sigma_\sigma \subseteq \mathcal{A}^* \times T_\sigma \times \mathbb{N}$ is the set of all attribute certificates attested by σ with the corresponding deadline. As on commented above, elements of Σ_σ are denoted as $\sigma\langle\langle a, t\rangle\rangle_d$. Sometimes, we will write them as $\langle\langle a, t\rangle\rangle_d$ for the sake of simplicity. We assume that SOAs only sign true certificates, that is, the following assertion holds:

$$\langle\langle a, t\rangle\rangle_d \in \Sigma_\sigma \Rightarrow \forall m \in \mathbb{N}.(ctime \leq m \leq d \Rightarrow a \in K(m)(t)) \tag{4.3}$$

3. Let $\mathcal{C}_\sigma = \mathcal{D}_\sigma \times \mathcal{A}$ be the set of certificate classes regarding SOA σ. Then, $SOAD_\sigma \subseteq \wp(\mathcal{C}_\sigma) \times OpSet \times \mathcal{A}^*$ is the SOA description constituted by a set of rules, each one representing a relation between a set of certificate classes and a given certificate class. The set of relational operators considered is $OpSet = \{\rightarrow, \varPhi\}$, where $\rightarrow$ is the usual implication, and operator $\varPhi$ is used to denote *inconsistency* and it will be formally defined below. For example, assuming that $\tau \in \mathcal{D}_\sigma$, rule $\tau\langle\langle b\rangle\rangle, \sigma\langle\langle c\rangle\rangle \rightarrow \sigma\langle\langle a\rangle\rangle$ could be an element of $SOAD_\sigma$, indicating that any target holding attributes b and c also holds a. In addition, the rule also expresses that σ delegates the task of checking b to SOA τ. It is worth noting that certificate classes appearing at the right side of rules always refer to the SOA defining the rule, this is why no SOA identifier is needed and it will be omitted for the sake of simplicity. As before, we assume that SOAD rules only establish true relations among attributes, that is, the following assertion holds

$$\begin{aligned} \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle a\rangle\rangle \in SOAD_\sigma \Rightarrow \\ \forall m \in \mathbb{N}, \forall t \in T_\sigma.(\{a_1, \cdots, a_n\} \subseteq K(m)(t) \Rightarrow a \in K(m)(t)) \end{aligned} \tag{4.4}$$

The Semantic Access Control (SAC) makes use of SOAD rules to derive information about properties. We have developed two derivation relations, $\vdash_r^\sigma$ (Figure 1) deduces information from certificate classes, and $\vdash_{at}^\sigma$ (Figure 2) deals with attribute certificates. In order to avoid confusion, from now on, we call d-rules the rules appearing in these two figures.

In the d-rules appearing in the figures, we are assuming that σ is the SOA from which new rules or attribute certificates are being inferred. The rest of the SOAs are supposed to belong to $\mathcal{D}_\sigma$.

Next, we give short explanations about the meaning of each d-rule of Figure 1.

R1 Every $SOAD_\sigma$ rule is directly derived by $\vdash_r^\sigma$. Note that the left part of each rule may contain references to other SOAs meaning delegation for checking the corresponding attribute. Since $\sigma \in \mathcal{D}_\sigma$, **R1** also deals with rules of the form $\sigma\langle\langle a\rangle\rangle \rightarrow \langle\langle b\rangle\rangle$.

R2 This d-rule defines transitivity. It may be directly inferred using the d-rules for $\vdash_{at}^\sigma$ described below. However, we define it explicitly in order to simplify the algorithms implementing the SAC deductive system.

R3 This d-rule defines the inconsistency between a given certificate class issued by σ and a set of them. If we can deduce $\sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle !b\rangle\rangle$ then

$$\textbf{R1}\ \frac{\sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle b\rangle\rangle \in SOAD_\sigma}{\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle b\rangle\rangle} \qquad (SOAD_\sigma \text{ rules})$$

$$\textbf{R2}\ \frac{\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle a\rangle\rangle, \vdash_r^\sigma \sigma\langle\langle a\rangle\rangle \rightarrow \langle\langle b\rangle\rangle}{\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle b\rangle\rangle} \qquad \text{(Transitivity)}$$

$$\textbf{R3}\ \frac{\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle !b\rangle\rangle}{\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \varPhi \langle\langle b\rangle\rangle} \qquad \text{(Inconsistency)}$$

$$\textbf{R4}\ \frac{\vdash_r^\tau \sigma\langle\langle b\rangle\rangle \rightarrow \langle\langle !a\rangle\rangle}{\vdash_r^\sigma \tau\langle\langle a\rangle\rangle \rightarrow \langle\langle !b\rangle\rangle} \qquad \text{(Exclusion)}$$

Fig. 1. d-rules for certificate classes

we conclude that attributes $\sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle$ and $\langle\langle b\rangle\rangle$ are inconsistent, that is, they cannot be held simultaneously. We use symbol $\varPhi$ to denote inconsistency.

R4 The exclusion d-rule says that both $\tau\langle\langle b\rangle\rangle \rightarrow \langle\langle !a\rangle\rangle$ in SOAD$_\sigma$ and $\sigma\langle\langle a\rangle\rangle \rightarrow \langle\langle !b\rangle\rangle$ in SOAD$_\tau$ may be used to prove that attributes a and b are inconsistent. Note that although these two assertions are logically equivalent, it is possible that we can prove only one of them, depending on the rules appearing in the corresponding SOADs. This d-rule also includes the case where SOAs σ and τ coincide.

The following proposition proves that $\vdash_r^\sigma$ only produces true relations among certificate classes. See appendix for proofs of this section.

Proposition 1. *If* $\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle c\rangle\rangle$ *then* $\forall m \in \mathbb{N}, t \in T_\sigma$*, if* $\{a_1, \cdots, a_n\} \subseteq K(m)(t) \Rightarrow c \in K(m)(t)$*.*

Note that the previous proposition does not consider operator $\varPhi$ because it only provides a specific notation for the rules having on their right side attributes of the form $!b$.

Definition 1 (Consistency). *We say that certificate class* $\sigma\langle\langle a\rangle\rangle \in \mathcal{C}_\sigma$ *is consistent with the certificate classes* $\sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma\langle\langle a_n\rangle\rangle \in \mathcal{C}_\sigma$*, and denote it as* $\sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma\langle\langle a_n\rangle\rangle \copyright \sigma\langle\langle a\rangle\rangle$*, iff* $\nvdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \varPhi \langle\langle a\rangle\rangle$*.*

It is worth noting that Proposition 1 implies that if SOA σ proves inconsistency then we have assured that the corresponding attributes cannot hold simultaneously. On the other hand, the fact that SOA σ cannot prove inconsistency does not necessarily imply that attributes are consistent. That is, the notion of *consistency* is weaker than that of *inconsistency*.

Figure 2 shows the system derivation used by a given SOA σ to infer certificate classes. We have denoted this relation with $\vdash_{at}^\sigma$.

In the following, we briefly explain the derivation rules given in the figure.

$$\textbf{A1}\ \frac{\langle\langle a,t\rangle\rangle_d \in \Sigma_\sigma, ctime \leq d}{\vdash^{\sigma}_{at} \langle\langle a,t\rangle\rangle_d} \qquad \text{(SOA At. Certif.)}$$

$$\textbf{A2}\ \frac{\begin{array}{l}\vdash^{\sigma}_{r} \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle a\rangle\rangle, \\ \forall 1 \leq i \leq n.(\vdash^{\sigma_i}_{at} \langle\langle a_i,t\rangle\rangle_{d_i}, ctime \leq d_i), d = min(d_1, \cdots, d_n)\end{array}}{\vdash^{\sigma}_{at} \langle\langle a,t\rangle\rangle_d} \qquad \text{(Rule Application)}$$

$$\textbf{A3}\ \frac{\begin{array}{l}\vdash^{\sigma}_{r} \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \Phi \langle\langle a\rangle\rangle, \\ \forall 1 \leq i \leq n, (\vdash^{\sigma_i}_{at} \langle\langle a_i,t\rangle\rangle_{d_i}, ctime \leq d_i), d = min(d_1, \cdots, d_n)\end{array}}{\vdash^{\sigma}_{at} \langle\langle !a,t\rangle\rangle_d} \qquad \text{(Inconsistency)}$$

Fig. 2. d-rules for attribute certificates

A1 Non expired attribute certificates in Σ_σ are directly inferred by $\vdash^{\sigma}_{at}$.

A2 Given a rule deduced using $\vdash^{\sigma}_{r}$, if each SOA σ_i asserts that target t holds attribute a_i, and the corresponding deadline d_i has not been reached, then σ derives the attribute certificate $\langle\langle a,t\rangle\rangle_d$, d being the minimum of the deadlines d_i.

A3 This d-rule simply applies the d-rule for inconsistency **R3**.

The following theorem proves the correctness of the information provided by SOAs. In summary, it establishes that each certificate issued is true.

Theorem 1 (Soundness). *For each attribute $a \in \mathcal{A}^*$ and target $t \in T_\sigma$, if a SOA σ exists such that $\vdash^{\sigma}_{at} \langle\langle a,t\rangle\rangle_d$ then $\forall m.ctime \leq m \leq d$, $a \in K(m)(t)$, that is, SOAs only certify true attribute certificates.*

4.2 Dealing with negation

In the previous section we have managed three types of negation: "$\neg$", "!" and "$\nvdash^{\sigma}_{at}$". In this section, we clarify the relations among them, and their effect when a particular SOA σ must issue certificates.

Definition 2. *We say that $\sigma \in \mathcal{S}$ does not issue an attribute certificate $\langle\langle a,t\rangle\rangle_d$ and denote it as $\vdash^{\sigma}_{at} \neg\langle\langle a,t\rangle\rangle_d$ iff $\nvdash^{\sigma}_{at} \langle\langle a,t\rangle\rangle_d$.*

Observe that $\vdash^{\sigma}_{at} \neg\langle\langle a,t\rangle\rangle_d$ has the effect of denying target t the access to resources if attribute a is necessary. However, this refutation may be produced due to very different motives, as commented on below.

Expression $\vdash^{\sigma}_{at} \neg\langle\langle a,t\rangle\rangle_d$ means that SOA σ cannot assert $\langle\langle a,t\rangle\rangle_d$. This may occur when $\langle\langle a,t\rangle\rangle_d$ cannot be deduced because there is no sufficient information in SOAs to assure it. However this situation may also take place if σ may deduce the opposite attribute, i. e., if $\vdash^{\sigma}_{at} \langle\langle \neg a,t\rangle\rangle_d$. That is, $\vdash^{\sigma}_{at} \neg\langle\langle a,t\rangle\rangle_d$ is weaker than $\vdash^{\sigma}_{at} \langle\langle \neg a,t\rangle\rangle_d$. It may be that σ cannot derive $\langle\langle a,t\rangle\rangle_d$, even although the assertion is true, that is, $\forall m. ctime \leq m \leq d, a \in K(m)(t)$, or equivalently, target t does hold a until time instant d.

On the other hand, note that $\langle\langle !a, t\rangle\rangle_d$ is different from $\langle\langle \neg a, t\rangle\rangle_d$. The first expression assures that target t does not hold a until time d, while the second one says that t holds $\neg a$ until time d.

From the user point of view, as commented on above, the three negations imply that t is not allowed to access a given resource, if attribute a is necessary.

We now formalize the relations among these three types of negations.

Using condition (4.2), it is sound to add the d-rules of Figure 1 to Figure 3.

$$\boxed{\textbf{R0}\ \frac{\forall \tau\langle\langle \neg a\rangle\rangle \in \mathcal{C}_\sigma}{\vdash_r^\sigma \tau\langle\langle \neg a\rangle\rangle \rightarrow \langle\langle !a\rangle\rangle}\ \text{(Negation)}}$$

Fig. 3. New d-rules for certificate classes

The following proposition shows how "$\neg$", "!" and "$\nvdash_{at}^\sigma$" are related.

Proposition 2. $\forall \sigma \in \mathcal{S}, t \in T_\sigma, m \in \mathbb{N}$,

$$\vdash_{at}^\sigma \langle\langle \neg a, t\rangle\rangle_m \Rightarrow \vdash_{at}^\sigma \langle\langle !a, t\rangle\rangle_m \Rightarrow \vdash_{at}^\sigma \neg\langle\langle a, t\rangle\rangle_m \tag{4.5}$$

5 Conclusions and further work

The SAC model has proven to be scalable and applicable to different environments with heterogeneous and complex access criteria. Moreover, other access control models can be represented within SAC. An infrastructure implementing this access control model called XSCD (XML-based Secure Content Distribution), complemented by autonomous enforcement mechanisms, has been developed and successfully applied to information commerce [12], digital rights management [13] and secure content distribution in digital libraries [22]. Another interesting application scenario for SAC is Web Services, where SAC achieved the desired semantic interoperability [21], and CORBA architecture [11].

The ability to perform a semantic validation of access control policies was an essential design goal of the SAC model. Both the Semantic Policy Language (SPL) defined in SAC and the semantic descriptions of the certificates issued by each SOA (conveyed by SOAD documents) were designed to serve this objective. The semantic validation ensures that the policies written by the security administrator produce the desired effects. In this paper, we have presented the semantic validation algorithms for access control policies developed as part of SAC. Additionally, the SAC model has been formalized and some important features have been formally proved. More specifically, the inference rules for deducting new information have been presented as part of this formal model, providing proofs of the correctness of SAC inference rules.

Regarding future work, model checking of the semantic validation algorithms will be developed in the near future. On the other hand, we are now working on the extension of the Semantic Policy Language with additional digital rights specification, along with semantic models for its management.

References

1. A. Baraani, J. Pieprzyk, and R. Safavi-Naini. Security In Databases: A Survey Study, 1996.
2. E. Bertino, S. Castano, and E. Ferrari. Securing XML documents with Author-X. *IEEE Internet Computing*, 5(3):21–31, May/June 2001.
3. M. Blaze, J. Feigenbaum, J. Ioannidis, and A.D. Keromytis. The role of trust management in distributed systems security. *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*, pages 185–210, 1993.
4. M. Blaze, J. Feigenbaum, and J. Lacy. Decentralized Trust Management. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 164–173, 1996.
5. D. W. Chadwick and A. Otenko. The PERMIS X.509 role based privilege management infrastructure. *Future Generation Computer Systems*, 19(2):277–289, 2003.
6. E. Damiani, S. de Capitani di Vimercati, S. Paraboschi, and P. Samarati. A Fine-Grained Access Control System for XML Documents. *In ACM Transactions on Information and System Security (TISSEC)*, 5(2):169–202, May 2002.
7. M. Thompson et al. Certificate-based Access Control for Widely Distributed Resources. In *Proc. of the 8th USENIX Security Symposium*, pages 215–227, 1999.
8. ITU-T. Recommendation X.509: Information Technology - Open Systems Interconnection - The Directory: Public-key and attribute certificate frameworks. http://www.itu.int/rec/recommendation.asp?type=folders&lang=e&parent=t-rec-x.509, 2000.
9. S. Jajodia, P. Samarati, and V.S. Subrahmanian. A Logical Language for Expressing Authorizations. In *Proc. of the IEEE Symposium on Security and Privacy*, pages 31–42, 1997.
10. M. Kudo and S. Hada. XML Document Security based on Provisional Authorisation. In *Proc. of the 7th ACM Conference on Computer and Communications Security*, pages 87–96, 2000.
11. J. López, A. Maña, J.J.Ortega, J.M. Troya, and M.I. Yagüe. Integrating PMI services in CORBA Applications. *Computer Standards and Interfaces Journal*, 25(4):391–409, 2003.
12. A. Maña, M.I. Yagüe, and V. Benjumea. Ec-gate: Electronic commerce based on e-gate technology, Golden Award of EGATE Open Contest 2002, paris, november 2002.
13. A. Maña, M.I. Yagüe, and V. Benjumea. EC-GATE: An Infrastructure for DRM. In *Proc. of the IASTED Intl. Conference on Communication, Network, and Information Security*, pages 283–288, 2003.
14. OASIS. XACML 1.1 Specification Set, 2003.
15. X. Qian and T.F. Lunt. A MAC Policy Framework for Multilevel Relational Databases. *IEEE Transactions on Knowledge and Data Engineering*, 8(1):1–14, February 1996.
16. P. Samarati and S. de Capitani di Vimercati. Access Control: Policies, Models, and Mechanisms. In *FOSAD 2000*, volume 2171 of *Lecture Notes in Computer Science*, pages 137–196. Springer, 2001.

17. R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-Based Access control Models. *IEEE Computer*, 29(2):38–47, 1996.
18. T. Sundsted. With Liberty and single sign-on for all. The Liberty Alliance Project seeks to solve the current online identity crisis, 2002.
19. T.Y.C. Woo and S.S. Lam. Authorizations in distributed systems: A new approach. *Journal of Computer Security*, 2(2):107–136, 1993.
20. M.I. Yagüe. *Modelo basado en Metadatos para la Integración Semántica en Entornos Distribuidos. Aplicación al Escenario de Control de Accesos*. Ph.D. dissertation, Computer Science Department. University of Málaga, 2003.
21. M.I. Yagüe, A. Maña, and J. López. A Metadata-based Access Control Model for Web Services. *Internet Research Journal: Electronic Networking Applications and Policy*, 25(1):99–116, 2005.
22. M.I. Yagüe, A. Maña, J. López, J. Pimentel, and J.M. Troya. A Secure Solution for Commercial Digital Libraries. *Online Information Review Journal*, 27(3):147–159, 2003.
23. M.I. Yagüe, A. Maña, J. López, and J.M. Troya. Applying the Semantic Web Layers to Access Control. In *Proc. of the Int. Workshop on Web Semantics*, pages 47–63. IEEE Computer Society Press, September 2003.
24. M.I. Yagüe and J.M. Troya. A Semantic Approach for Access Control in Web Services. In *Proc. of the W3C Euroweb 2002 International Conference*, 2002.

A Example

To illustrate the inference rules stated on SOAD documents as the basis for the semantic validation of policies, let us consider an editorial and its digital library composed of books, magazines, bulletin news and other relevant publications. The editorial has some special discounts for some customers; and also privileged customers who can freely access some types of resources. For example, the University of Málaga has a particular membership with this editorial which grants some privileges to their staff.

The access control system is based on the Semantic Access Control model, and hence we have the separated specifications of PAS, SRR, and Policy to describe access control criteria. Figure 4 is the XML representation of the semantic properties relevant to access to the Computer News magazine. Figure 5.a shows a simple policy (FreeDownload.xml) that defines as access criteria to be holder of an attribute certificate signed by the SOA of the editorial attesting the subscription to the editorial Portal. Figure 5.b represents the Policy Allocation Specification document which allocates the FreeDownload.xml policy to magazine items accessible in the digital library through the portal.

When a user tries to access the Computer News magazine through the McGrow portal, thanks to the semantic information represented in the Secured Resource Representation for this object (Figure 4), dynamic allocation is made on the basis of PAS of Figure 5.b. Therefore, the policy of Figure 5.a is used to control the access to this object, based on its semantic property of being a magazine.

As in any access control scheme based on attribute certificates, the semantics of policies in the SAC model heavily depend on the semantics of attribute

```
<?xml version=“1.0” encoding=“UTF-8”?>
<SRR xmlns=“http://www.lcc.uma.es/∼yague”
xmlns:xsi=”http://www.w3.org/2001/XMLSchema-instance”
xsi:schemaLocation=”http://www.lcc.uma.es/ yague SRR.xsd”
Resource=”http://www.mcgrow.com/”>
        <Property>
            <PropertyName>PublicationName</PropertyName>
            <PropertyValue>Computer_News</PropertyValue>
        </Property>
        <Property>
            <PropertyName>PublicationType</PropertyName>
            <PropertyValue>magazine</PropertyValue>
        </Property>
</SRR>
```

Fig. 4. SRR for the Computer_News magazine

```
<?xml version=“1.0” encoding=“UTF-8”?>
<Policy xmlns=“http://www.lcc.uma.es/∼yague”
xmlns:xsi=
“http://www.w3.org/2001/XMLSchema-instance”
xsi:schemaLocation=
“http://www.lcc.uma.es/∼yague Policy.xsd”>
<AccessRules>
    <AccessRule>
        <AttributeSet>
            <Attribute>
                <AttributeName>Subscription
                </AttributeName>
                <AttributeValue>Portal
                </AttributeValue>
                <SOA_ID>McGrow_SOA</SOA_ID>
            </Attribute>
        </AttributeSet>
    </AccessRule>
</AccessRules>
</Policy>
```

```
<?xml version=“1.0” encoding=“UTF-8”?>
<PAS xmlns=
“http://www.lcc.uma.es/∼yague”
xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
xsi:schemaLocation=
“http://www.lcc.uma.es/∼yague pas.xsd”>
<Policy>FreeDownload.xml</Policy>
<Object>
 <ObjectLocation>http://www.mcgrow.com/portal/
 </ObjectLocation>
 <Conditions>
     <Condition>
         <PropertyName>PublicationType
         </PropertyName>
         <PropertyValue>magazine
         </PropertyValue>
     </Condition>
 </Conditions>
</Object>
</PAS>
```

Fig. 5. (a) FreeDownload.xml policy (b) PAS for magazines

certificates which we have modelled in SOAD (Source of Authorization Description) documents. Figure 6 shows the descriptions of the source of authorization that certifies the membership to the University of Malaga (UMA). The rule of this SOAD states that the Source Of Authorization (SOA) of UMA trusts in the SOA of the Computer Science Department for attesting to membership to the department. That is, in order to prove UMA membership, to present an attribute certificate signed by the CS department SOA attesting to being a member of this department will be equivalent to presenting an attribute certificate signed by the UMA attesting to membership of the UMA. Figure 7 shows the descriptions of the source of authorization corresponding to the McGrow editorial. Relations among attributes certified by each SOA are also described in these documents.

The SOAD corresponding to the McGrow editorial has two rules. The first rule states that to be a member of UMA with a trusted certificate from UMA_SOA implies being a McGrow special customer and, additionally, being subscribed to its portal. The second rule states that a certificate of being subscribed to the portal implies a certificate of subscription to the Computer_News and Math_News magazines.

To see the important role of the inference mechanisms developed, we consider a professor of UMA who wants to access one of the CS_News magazines.

```
<?xml version=“1.0” encoding=“UTF-8”?>
<SOAD xmlns:xsi=“http://www.w3.org/2001/XMLSchema-instance”
xsi:noNamespaceSchemaLocation=“SOAD.xsd”>
  <SOA_ID>UMA_SOA</SOA_ID>
  <ACDeclarations>
    <SOAAttribute>
    <AttributeName>Member</AttributeName>
    <AttributeValue>UMA</AttributeValue>
    </SOAAttribute>
  </ACDeclarations>
  <ACRelations>
    <SOARule>
      <AttributeSet>
      <SOAAttribute>
        <AttributeName>Member</AttributeName>
        <AttributeValue>CSDepartment</AttributeValue>
        <SOA_ID>CSDpt_SOA<SOA_ID>
      </SOAAttribute>
    </AttributeSet>
    <Relation>Implies</Relation>
    <AttributeSet>
      <SOAAttribute>
        <AttributeName>Member</AttributeName>
        <AttributeValue>UMA</AttributeValue>
      </SOAAttribute>
    </AttributeSet>
    </SOARule>
  </ACRelations>
</SOAD>
```

Fig. 6. SOAD of the University of Málaga SOA

If this professor presents an attribute certificate signed by the Computer Science department SOA stating he/she is a professor of this University then this certificate will be equivalent to an attribute certificate signed by the McGrow Editorial of being subscribed to the portal. Therefore, the policy requisites stated on the access control policy will be satisfied and he/she will get free access to this document.

Finally, let us consider derivation rules stated in SOAD documents and how information from certificate classes is deduced. Let σ_{mg}, σ_{uma} and σ_{cs} be the SOAs for the McGrow Editorial, the University of Málaga and the Computer Science Department. Figure 8 shows the rules included in the Source Of Authorization Description (SOAD) documents of MacGrow and UMA. In SOAD$_{\sigma_{mg}}$, the first rule states that to be member of UMA implies to be a Privileged Customer of MacGrow. Second rule states that to be a member of UMA implies being subscribed to the MacGrow Portal. Last two rules state that to be subscribed to this portal implies being subscribed to the Computer News and Math News magazines, respectively. The only rule of SOAD $_{\sigma_{uma}}$ states that to be a member of the Computer Science Department implies being a member of the University of Malaga. Now, suppose that $\vdash_{at}^{\sigma_{cs}} \sigma_{cs}\langle\langle \mathtt{MemCS}, \mathtt{MYagüe}\rangle\rangle$, that is, the Computer Science Department is able to certify that `MYagüe` is a member of the Department.

The following derivation shows how SOA σ_{mg} infers she can access the Computer News magazine, using the d-rules described above. For the sake of simplicity, we have dropped the time parameter, assuming that attribute certificates are always valid. We have divided the derivation into three parts. The last derivation makes use of the results previously obtained to reach the conclusion. Note that the rule applied appears at the left side of each derivation.

```
<?xml version="1.0" encoding="UTF-8"?>
<SOAD xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="SOAD.xsd">
<SOA_ID>McGrow_SOA</SOA_ID>
<ACDeclarations>
    <SOAAttribute>
        <AttributeName>Suscription
        </AttributeName>
        <AttributeValue>McGrow_Portal
        </AttributeValue>
    </SOAAttribute>
    <SOAAttribute>
        <AttributeName>Subscription
        </AttributeName>
        <AttributeValue>Computer_News
        </AttributeValue>
    </SOAAttribute>
    <SOAAttribute>
        <AttributeName>Subscription
        </AttributeName>
        <AttributeValue>Math_News
        </AttributeValue>
    </SOAAttribute>
    <SOAAttribute>
        <AttributeName>Customer
        </AttributeName>
        <AttributeValue>Privileged
        </AttributeValue>
    </SOAAttribute>
</ACDeclarations>
```

```
/* being member of UMA implies to be a customer with
some privileges and subscription to the editorial portal */
<ACRelations>
<SOARule>
    <AttributeSet>
        <SOAAttribute>
        <AttributeName>Member</AttributeName>
        <AttributeValue>UMA</AttributeValue>
        <SOA_ID>UMA_SOA</SOA_ID>
        </SOAAttribute>
    </AttributeSet>
    <Relation>Implies</Relation>
    <AttributeSet>
        <SOAAttribute>
        <AttributeName>Customer</AttributeName>
        <AttributeValue>Privileged</AttributeValue>
        <SOA_ID>MacGrow_SOA</SOA_ID>
        </SOAAttribute>
        <SOAAttribute>
        <AttributeName>Subscription</AttributeName>
      <AttributeValue>McGrow_Portal </AttributeValue>
        <SOA_ID>MacGrow_SOA</SOA_ID>
        </SOAAttribute>
    </AttributeSet>
</SOARule>
</ACRelations>

<ACRelations>
<SOARule>
    <AttributeSet>
        <SOAAttribute>
        <AttributeName>Suscription</AttributeName>
        <AttributeValue>McGrow_Portal>/AttributeValue>
        </SOAAttribute>
    </AttributeSet>
    <Relation>Implies</Relation>
    <AttributeSet>
        <SOAAttribute>
        <AttributeName>Subscription</AttributeName>
        <AttributeValue>Computer_News</AttributeValue>
        </SOAAttribute>
    <AttributeSet>
        <SOAAttribute>
        <AttributeName>Subscription</AttributeName>
        <AttributeValue>Math_News</AttributeValue>
        </SOAAttribute>
    </AttributeSet>
</SOARule>
</ACRelations>
</SOAD>
```

Fig. 7. SOAD of the McGrow Editorial SOA

$$(\mathbf{A2})\ \frac{\vdash^{\sigma_{cs}}_{at} \langle\langle \texttt{MemCS}, \texttt{MYagüe}\rangle\rangle \quad (\mathbf{R1})\ \dfrac{\sigma_{cs}\langle\langle \texttt{MemCS}\rangle\rangle \longrightarrow \langle\langle \texttt{MemUma}\rangle\rangle \in SOAD_{\sigma_{uma}}}{\vdash^{\sigma_{uma}}_{r}\ \sigma_{cs}\langle\langle \texttt{MemCS}\rangle\rangle \longrightarrow \langle\langle \texttt{MemUma}\rangle\rangle}}{\vdash^{\sigma_{uma}}_{at}\ \langle\langle \texttt{MemUma}, \texttt{MYagüe}\rangle\rangle}$$

$$(\mathbf{R1})\ \frac{\sigma_{uma}\langle\langle \texttt{MemUma}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_MGPortal}\rangle\rangle \in SOAD_{\sigma_{mg}}}{\vdash^{\sigma_{mg}}_{r}\ \sigma_{uma}\langle\langle \texttt{MemUma}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_MGPortal}\rangle\rangle}$$

$$(\mathbf{R1})\ \frac{\sigma_{mg}\langle\langle \texttt{S\_MGPortal}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_CNews}\rangle\rangle \in SOAD_{\sigma_{mg}}}{\vdash^{\sigma_{mg}}_{r}\ \sigma_{mg}\langle\langle \texttt{S\_MGPortal}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_CNews}\rangle\rangle}$$

$$(\mathbf{R2})\ \frac{\vdash^{\sigma_{mg}}_{r}\ \sigma_{uma}\langle\langle \texttt{MemUma}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_MGPortal}\rangle\rangle \ \vdash^{\sigma_{mg}}_{r}\ \sigma_{mg}\langle\langle \texttt{S\_MGPortal}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_CNews}\rangle\rangle}{\vdash^{\sigma_{mg}}_{r}\ \sigma_{uma}\langle\langle \texttt{MemUma}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_CNews}\rangle\rangle}$$

$$(\mathbf{A2})\ \frac{\vdash^{\sigma_{uma}}_{at}\ \langle\langle \texttt{MemUma}, \texttt{MYagüe}\rangle\rangle \ \vdash^{\sigma_{mg}}_{r}\ \sigma_{uma}\langle\langle \texttt{MemUma}\rangle\rangle \longrightarrow \langle\langle \texttt{S\_CNews}\rangle\rangle}{\vdash^{\sigma_{mg}}_{at}\ \langle\langle \texttt{S\_CNews}, \texttt{MYagüe}\rangle\rangle}$$

$\sigma_{uma}\langle\langle\texttt{MemUma}\rangle\rangle \rightarrow \langle\langle\texttt{CPriv}\rangle\rangle$ $\sigma_{uma}\langle\langle\texttt{MemUma}\rangle\rangle \rightarrow \langle\langle\texttt{S\_MGPortal}\rangle\rangle$ $\sigma_{mg}\langle\langle\texttt{S\_MGPortal}\rangle\rangle \rightarrow \langle\langle\texttt{S\_CNews}\rangle\rangle$ $\sigma_{mg}\langle\langle\texttt{S\_MGPortal}\rangle\rangle \rightarrow \langle\langle\texttt{S\_MNews}\rangle\rangle$	$\sigma_{cs}\langle\langle\texttt{MemCS}\rangle\rangle \rightarrow \langle\langle\texttt{MemUma}\rangle\rangle$

Fig. 8. Rules in $\mathrm{SOAD}_{\sigma_{mg}}$ and $\mathrm{SOAD}_{\sigma_{uma}}$

B Proofs

Proposition 1 If $\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle c\rangle\rangle$ then $\forall m \in \mathbb{N}, t \in T_\sigma$, if $\{a_1, \cdots, a_n\} \subseteq K(m)(t) \Rightarrow c \in K(m)(t)$.

Proof. Denote $R \equiv \vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle c\rangle\rangle$, and consider $t \in T_\sigma$ and $m \in \mathbb{N}$ such that $\{a_1, \cdots, a_n\} \subseteq K(m)(t)$. Now, we reason by induction on the depth of the derivation tree to produce R.

- Base case. If $\sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle c\rangle\rangle \in SOAD_\sigma$, the proof is directly derived from condition (4.4).
- Inductive case. We have two possible cases:
  1. If R has been obtained applying rule **R2**, there exists $d \in \mathcal{A}$ and two rules such that $\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle d\rangle\rangle$, and $\vdash_r^\sigma \sigma\langle\langle d\rangle\rangle \rightarrow \langle\langle c\rangle\rangle$. Applying successively the induction hypothesis to these rules, firstly we deduce $d \in K(m)(t)$, and then $c \in K(m)(t)$.
  2. If R has been obtained applying rule **R4**, then $\exists b \in \mathcal{A}.c =!b$ and $R \equiv \vdash_r^\sigma \tau\langle\langle a\rangle\rangle \rightarrow \langle\langle !b\rangle\rangle$. Applying the induction hypothesis to $\vdash_r^\tau \sigma\langle\langle b\rangle\rangle \rightarrow \langle\langle !a\rangle\rangle$, we have that if $b \in K(m)(t)$ then $a \notin K(m)(t)$. Thus, assuming that $a \in K(m)(t)$, we deduce that $b \notin K(m)(t)$, or equivalently by definition (condition (4.1)) that $c =!b \in K(m)(t)$.

Theorem 1 For each attribute $a \in \mathcal{A}^*$ and target $t \in T_\sigma$, if a SOA σ exists such that $\vdash_{at}^\sigma \langle\langle a, t\rangle\rangle_d$ then $\forall m.ctime \leq m \leq d$, $a \in K(m)(t)$, that is, SOAs only certify true attribute certificates.

Proof. By induction on the depth of the derivation tree to assert $\langle\langle a, t\rangle\rangle_d$.

- If $\langle\langle a, t\rangle\rangle_d \in \Sigma_\sigma$ then, by condition (4.3), we have that $\forall m.ctime \leq m \leq d, a \in K(m)(t)$.
- Let us assume that we have applied rule **A2** to deduce $\langle\langle a, t\rangle\rangle_d$. Consider an index $i(1 \leq i \leq n)$. By induction hypothesis, if $\vdash_{at}^{\sigma_i} \langle\langle a_i, t\rangle\rangle_{d_i}, ctime \leq d_i$ then $\forall m_i.ctime \leq m_i \leq d_i, a_i \in K(m_i)(t)$. Thus, defining $d = min(d_1, \cdots, d_n)$, we deduce that $\forall m.ctime \leq m \leq d, \{a_1, \cdots, a_n\} \subseteq K(m)(t)$. Finally, applying Proposition 1 to $\vdash_r^\sigma \sigma_1\langle\langle a_1\rangle\rangle, \cdots, \sigma_n\langle\langle a_n\rangle\rangle \rightarrow \langle\langle a\rangle\rangle$, it is derived $\forall m.ctime \leq m \leq d, a \in K(m)(t)$.
- The proof corresponding to applying **A3** in the derivation is similar to the previous one.