Credential Based Access Control for Semantic Web Services

Sudhir Agarwal
Institute of Applied Informatics and
Formal Description Methods (AIFB),
University of Karlsruhe (TH), Germany.
agarwal@aifb.uni-karlsruhe.de

Barbara Sprick and **Sandra Wortmann**
Information Systems and Security,
University of Dortmund, Germany.
{barbara.sprick,sandra.wortmann}@udo.edu

Abstract

In this paper we make a contribution to the proof and trust layer of the Semantic Web layer cake by integrating two well founded techniques, namely DAML-S (for describing Web services with machine-processable semantics) and SPKI/SDSI (for specifying authorization based access control). Our approach builds on the idea of autonomous granting of access rights and decision making based on independent trust structures. Our framework allows the specification of access control related and functionality related aspects in a unified way that is manageable and efficient. Therefore, our approach is useful not only in typical Web service based applications (client-server architecture) but also in peer to peer and agent based applications.

1 Introduction

With the advent of the Semantic Web (Berners-Lee, Hendler, & Lassila 2001; Fensel *et al.* 2002; Patel-Schneider & Fensel 2002), Web services have gained even more importance (Ankolekar *et al.* 2002). Semantic Web techniques, especially ontologies, allow to describe Web services with more machine understandable semantics, thus enabling new features like automatic composition, simulation and discovery of Web services (Ankolekar *et al.* 2002; Burstein *et al.* 2003). The vision of the Semantic Web is to make the current Web more like an information system. In such an information system Web services play the role of operations available to the users. However, the use of Web services is not restricted to access information, but also in many other areas, for example electronic business and enterprise application integration.

Because of the vast heterogeneity of the available information, information providers and users, security becomes extremely important. Security related aspects are mostly classified in three categories, namely confidentiality, integrity and availability (Bishop 2003; Samarati & Capitani di Vimercati 2001; Denning 1982). Access control, which means the users must fulfill certain conditions in order to access certain functionality plays an important role in all three fields. For example, a student must show her library card to borrow a book from the university library. In context of confidentiality, it means that a student has access to the information relevant to only her own library account and thus can not know which other students have borrowed which books. In context of integrity, it means that a student may not change or cause a change in information relevant to the library account of another student. In context of availability, access control helps to prevent denial of service attacks that can take place if the access is uncontrolled.

Current access control is mostly based on authentication, which requires central control (registration) and proof of identity. This identity based authentication leads to certain limitations regarding the spontaneity and privacy and hence not always desired. Therefore, we propose authorization based access control rather than authentication based access control of Web services. Authorization based access control also includes authentication, but here the authentication is based on public keys and not on identities.

1.1 Authorization based Access Control

Currently, access control is based on identity based authentication, which means that the users must be known to the provider, for example via registration.

However, since the Semantic Web is an open, distributed, decentralized, dynamic and interoperable environment, in which Web services must be offerable and usable by anyone spontaneously and dynamically and users do not always wish to disclose their identities, we believe, that security infrastructures that require registrations or any other central controlling components are not suitable in the Semantic Web. Therefore, we propose authorization based access control rather than authentication based access control of Web services.

Credentials are digitally signed documents, which can be transmitted by untrusted channels like the Web, see e.g., (Biskup & Karabulut 2002; Chaum 1985). Credentials assert a binding between a principal and some property. A principal represents a user and depending on the context identified by her public or secret key. The meaning of a stated property may be a granted capability for a service, an identity or a non-identifying characteristic of a user like e.g., a skill. For further related work refer to (Brands 2000; Blaze *et al.* 1999; Samarati 2002). The credential-based public key infrastructure SPKI/SDSI (Ellison *et al.* 1999a; 1999b) allows each principal to issue credentials. Unlike

other public key infrastructures, SPKI/SDSI requires no central certification authority. Thus, each Web service provider can issue and trust credentials independent of other service providers and may even define her own trust structure. A Web service provider, acting as a verifier, can locally and autonomously decide whether access to her service should be granted or not. Access decisions are based on the provider's interpretation of a user's capabilities or characteristics given by shown SPKI/SDSI certificates. Furthermore, users can request Web services spontaneously without registering themselves with the individual Web service providers. Therefore SPKI/SDSI credentials are more suitable than the classical authentication based systems for specifying access control policies in the Semantic Web .

1.2 Requirements of Access Control for Semantic Web Services

Web services are meant to offer certain functionalities that depend on the input parameters supplied by its users. Often input parameters must fulfill certain conditions in order to assure correct behaviour of a Web service. Access control ensures that only eligible users get access to a Web service.

The access control policy of a Web service is specified by the provider of the Web service description which is mostly identical with the provider of the Web service. An end user knowing some Web services may combine few of them in some way to solve a certain task at hand. Prior to executing such a combination or plan she must know whether she can fulfill the access control policy of the plan. Hence we identify the following requirements for specifying access control for Semantic Web services:

R1 The framework must allow an end user to *check* and *prove* her eligibility for a Web service or a combination of Web services.

Now consider a Web service that offers electricity contracts and requires that the customer is at least 18 years of age. This requirement can be specified as access control policy of the Web service rather easily. However, the access control policies of most of the Web services are not so simple. For example, it is quite realistic that an electricity company offering such a Web service requires that the customer is at least 18 years of age as well as lives in a particular geographical region. The access control policy becomes even more complex when the access control requires not only that a user must have certain properties but also that a user may *not* have certain properties. For example, the customer may not have any outstanding accounts with the electricity company. We identify further requirements for specifying access control for Semantic Web services:

R2 The framework must support the specification of complex access control requirements.

Now consider that the electricity selling Web service has two input parameters, namely `deliveryAddress` and `noticePeriod`. The "functional" precondition for the `deliveryAddress` is that it must be a valid address in Germany and for `noticePeriod` is that it must be either 1 month or 3 months. Further, the Web service's access control policy requires that contracts with one month notice period and delivery address outside a particular geographical region are closed only with users who can prove their Greenpeace membership. Hence, we see that the access control requirements of a Web service may depend on the requested functionality (controlled by the values of the input parameters) and that the provided functionality may depend on the access control conditions fulfilled by the requester. Thus we identify that access control and functional aspects are not always independent of each other and consequently following requirement:

R3 The framework must be able to specify the interplay of the access control and functional aspects of the Web services.

Web services are typically distinguished in *atomic* and *composite* Web services. As the terms suggest, an atomic Web service is one that can not be further broken into parts, whereas a composite Web service is one that is decomposable into atomic and composite Web services, which are often referred to as component Web services. In addition to the set of component Web services, a composite Web service has a control flow and a data flow graph that contain information about how the component Web services are connected and how the data flows from one component Web service to another respectively.

Consider the following two Web services: (1) a Web service w_1 that offers Greenpeace membership and (2) our previous electricity selling Web service w_2 which requires Greenpeace membership for contracts with one month notice period for delivery addresses outside a particular geographical region. Now consider a composite Web service w_3 that first executes Web service w_1 and then Web service w_2, that is, it closes a Greenpeace membership before closing an electricity contract. Obviously, the access requirement "Greenpeace membership" of Web service w_2 is fulfilled after the execution of Web service w_1 and hence Greenpeace membership is not required to access the composite Web service w_3 although it is required by its component Web service w_2.

While the access control policy of an atomic Web service can be specified directly, the access control policy of a composite Web service depends on those of its component Web services and thus must be computed by the provider of the composite web service. Hence, we identify the following requirements for specifying access control for Semantic Web services:

R4 The framework must support a Web service provider in *computing* the access control policy of a composite Web service, whereby

R5 The framework must consider credentials that are issued to the requester on the fly, that is, during the execution of a composite Web service.

1.3 Our Approach

In this paper, we present a Semantic Web compatible approach for specifying access control policies of Web services to make Semantic Web services more useful while

keeping the spirit of the Semantic Web. We combine two well founded techniques, namely DAML-S (Ankolekar *et al.* 2002) for describing Web services and SPKI/SDSI for specifying access control policies. Our approach is one of the few contributions in the proof and trust layer of the Semantic Web layer cake (Patel-Schneider & Fensel 2002).

We view access control policies as conditions a Web service provider defines to restrict the set of users who may access the functionalities offered by her Web service. We introduce a policy algebra to specify and handle complex access control policies more efficiently. We show how a Web service provider can specify access control policies as policy algebra expressions in access control list entries and how access control lists can be integrated as preconditions in the description of a Web service.

A user wishing to access a functionality offered by a Web service must prove that she is eligible to gain access to the required functionality. On the basis of the access control policy of the Web service, the user calculates an appropriate subset of her credentials. She sends this subset of credentials to the Web service provider to prove her eligibility. The Web service provider acting as a verifier decides autonomously on the basis of the shown credentials and her trust structure whether access should be granted to the user or not. We will show, how the transmission of credentials can be made possible by augmenting the set of input parameters of a Web service by one parameter that carries the set of credentials from a user to a Web service provider.

This paper is structured as follows. In section 2, we give short introductions to the well known Semantic Web services description language, DAML-S and to the credential based public key infrastructure SPKI/SDSI. We introduce a policy algebra and show how it can be used in access control list entries to express access control policies more efficiently and dynamically. Then we present our main contribution by showing how SPKI/SDSI credentials can be integrated with DAML-S to specify access control policies and how users can interpret access control policies specified in a Web service description and act accordingly. In section 3, we present an application scenario and show by example how our framework can be used. Finally, we conclude in section 4.

1.4 Related Work

To the best of our knowledge, (Denker *et al.* 2003) is the only work that has dealt with the issue of security and DAML-S in detail. Our work is complementary to the mentioned contribution. (Denker *et al.* 2003) focus on developing security-related ontologies and two step matchmaking. Our focus is on a Semantic Web compliant security infrastructure and framework. We believe, that it is not always possible to handle security-related and functionality related characteristics separately because of the compositionality issues and anticipate a unified framework that can represent both types of properties. (Kagal, Finin, & Joshi 2003) introduce a policy language for marking up Semantic Web entities. However, it is not yet clear, how this policy language can be integrated and used with a Web service description language e.g., DAML-S.

2 Specification of Access Control Policies of Semantic Web Services

In our approach, we combine two well founded techniques, namely DAML-S (Ankolekar *et al.* 2002) for describing semantic Web services and SPKI/SDSI for specifying access control policies. In this section, we first give short introduction to DAML-S, SPKI/SDSI and policy algebra and then show how SPKI/SDSI can be integrated with DAML-S to enable access control with semantic Web services.

2.1 Introduction to DAML-S

DAML-S is a DAML+OIL ontology for describing Web services with the objective of making Web services computer-interpretable and hence enabling tasks like discovery, composition, simulation, interoperation and execution monitoring of Web services. DAML-S complements the various industrial efforts that are low-level, by providing Web service descriptions at application level (Ankolekar *et al.* 2002; Burstein *et al.* 2003). DAML-S has three main parts, namely `ServiceProfile`, `ServiceModel` and `ServiceGrounding`. `ServiceProfile` contains properties related to the functionality a service offers and answers the question *what does a service do?*, `ServiceModel` contains properties related to the operation of a Web service and answers the question *How does a service work?* and `ServiceGrounding` contains properties related to the access to a Web service and answers the question *How can a service be accessed?*

A service profile provides a high-level description of a service and its provider. It is used to request or advertise services with discovery services and capability registries. Service profiles consist of three types of information: a *description* of the service and the service provider; the *functional behavior* of the service and several *functional attributes* tailored for automated service selection.

The operation of a Web service is described in terms of a process model, which details both the control structure and data flow structure of the service. Two main components of the process model are the *process ontology*, which describes a service in terms of its inputs, output, preconditions, effects and where appropriate, its component subprocesses; and the *process control ontology* which describes each process in terms of its state, including initial activation, execution and completion. The primary kind of entity in the process ontology is *process*. DAML-S distinguishes between *atomic*, *simple* and *composite* processes. Atomic processes are directly invocable and execute in a single step. Simple processes, on the other hand, are not directly invocable and are not associated with a grounded. They are rather used as elements of abstraction. Composite processes are decomposable into other (non-composite or composite) processes. Their decompositions are specified by control constructs such as *sequence*, *if-then-else*, *repeat-while* etc..

A grounding can be thought of as a mapping from an abstract to concrete specification of those service description elements that are required for interacting with a service. The grounding of a service has mainly to do with the protocol

and message formats, serialization, transport and addressing.

In DAML-S, both `ServiceProfile` and `ServiceModel` are conceived as abstract representations whereas `ServiceGrounding` deals with the concrete level of specification. For more information on DAML-S, refer to (Ankolekar *et al.* 2002; Burstein *et al.* 2003).

2.2 Introduction to SPKI/SDSI

SPKI/SDSI is a credential based public key infrastructure resulted by merging SDSI (Simple Distributed Security Infrastructure) and SPKI (Simple Public Key Infrastructure). The SDSI part of SPKI/SDSI proposes the use of local names, the SPKI part deals with authorization and delegation of authorization. We consider SPKI/SDSI for credential based access control as proposed in (Ellison *et al.* 1999a; 1999b; Rivest & Lampson 1996).

The main advantage of SPKI/SDSI compared to other credential based systems is that it does not require central control and allows users, e.g., Web service providers to specify their own trust structures independent of each other. Each participant acting as certification authority (CA) can issue certificates to other users and acting as user can prove her eligibility by showing an appropriate set of credentials to the service provider. A Web service provider acting as a verifier checks the shown set of credentials against the access control policy of the Web service and grants or denies access accordingly.

SPKI/SDSI supports two kinds of credentials, namely *name certificates* to bind principals to names and *authorization certificates* to bind authorizations to names. Besides name certificates and authorization certificates, SPKI/SDSI also provides access control lists (ACL) for specifying access control policies for some interface.

Name Certificates An SPKI/SDSI name certificate is used to bind principals to a name and is a document of the form

$<$*Keyholder*,*Name*,*Subject*,*Validity*$>$

1 *Keyholder* represents the issuing principal who certifies the body with a signature.

2 *Name* is an identifier chosen by the issuing principal *Keyholder* to form a local name. In SPKI/SDSI every principal is associated with her local name space which is the set of her local names. A local name belonging to the name space of the principal *Keyholder* has the syntactical form *Keyholder Name* and is evaluated to a set of principals.

3 *Subject* inserts a principal, a local name or an extended name into the set of principals denoted by the local name *Keyholder Name*. Similar to local names, *Subject* is evaluated to a set of principals.

4 *Validity* denotes the validity of the certificate.

Consider for example a magazine company *EcoMag* that issues membership cards to its subscribers. A membership card for a subscriber *Alice* is modeled by the following SPKI/SDSI name certificate:

$<K_{\texttt{EcoMag}}$,`subscriber`,$K_{\texttt{Alice}}$,`till 2004_12_31`$>$.

In this certificate, $K_{\texttt{EcoMag}}$ denotes the public key of the magazine company *EcoMag* and $K_{\texttt{Alice}}$ denotes the public key of the subscriber *Alice*. By issuing this name certificate, the magazine company *EcoMag* certifies, that $K_{\texttt{Alice}}$ belongs to the local name $K_{\texttt{EcoMag}}$ `subscriber`, that is, $K_{\texttt{Alice}}$ belongs to the group (set) of subscribers of $K_{\texttt{EcoMag}}$. As stated in the validity field, this credential is valid until 31th December 2004.

Authorization Certificates An SPKI/SDSI authorization certificate is used to bind an authorization to a name and is a document of the form

$<$*Keyholder*,*Subject*,*Authorization*,*Delegation*,*Validity*$>$

that is signed by the keyholder.

1 *Keyholder* represents the issuing principal who certifies the body with the signature.

2 *Subject* denotes the set of grantees of the authorization, e.g., a principal or a local name.

3 *Authorization* specifies the granted permissions.

4 *Delegation* is a boolean flag, which, if set, means that the grantee is allowed to forward the permissions specified in *Authorization* to other principals.

5 *Validity* denotes the validity of the certificate.

Consider again the magazine company `EcoMag`. It grants full text access to the electronic edition of the magazine to its subscriber *Alice* by issuing the following authorization certificate: $<K_{\texttt{EcoMag}}$,K_{Alice},`fulltextaccess`,`false`,`till 2004_12_31`$>$.

Access Control Lists and Access Decision SPKI/SDSI provides access control lists (ACL) for specifying access control policies for some interface. An ACL is a list of ACL entries which are documents of the form

$<$`Self`,*Subject*,*Authorization*,*Delegation*,*Validity*$>$.

An ACL entry is equivalent to an authorization certificate except that it is not signed, the *Keyholder* is the reserved word `Self` instead of a key and it is not actually issued to a principal (i.e., it remains locally stored at the provider's site). Both authorization certificates and ACL entries can actually explicitly bind authorizations to principals, but, we focus on bindings between authorizations and names. Using such a binding is more efficient for specifying access control conditions since names denote sets of principals instead of single principals.

Consider again the previously mentioned magazine company *EcoMag* which grants full text access to the electronic edition of the magazine to all its subscribers. To specify this access control policy, it defines an ACL containing the following ACL entry:

$<$`Self`,$K_{\texttt{EcoMag}}$`subscriber`,`fulltextaccess`,.,.$>$.

To access an interface, a requester must prove her eligibility, that is, she must prove that her set of credentials fulfills the access control policy of the interface. To do so, she

constructs an authorizing set (chain) of certificates from the ACL and her set of certificates.

(Clarke *et al.* 2001) suggest a certificate chain discovery algorithm that constructs such a chain of certificates from a given set of certificates and an ACL. The algorithm interprets an ACL entry as an unsigned authorization certificate with `Self` as keyholder. It reduces chains of delegations and local name meanings in the set of credentials by computing the name reduction closure, by removing useless credentials (e.g., those that are not valid) and by constructing a graph from the remaining credentials. In this graph, each credential corresponds to a single directed edge that points from the certificate's issuer to its subject. The algorithm uses depth-first search to determine whether there is a path from `Self` to the requesting principal. In case of success, the principal sends the credentials lying on the determined path to the provider of the interface. The provider, acting as verifier, takes an access decision based on the transmitted set of credentials by constructing an unsigned authorization certificate of the form $<$`Self`,K_A,*Authorization*,.,.$>$.

2.3 Using Policy Algebra in Access Control List Entries

Web service providers specify access control policies of their Web services by defining ACLs. In section 1.2 we have identified that the framework must support the specification of complex access control requirements (**R2**) as well as a Web service provider in *computing* the access control policy of a composite Web service (**R4**). Thus the Web service provider needs a mechanism that allows to specify not only simple subjects (e.g., principals or local names) but also composed subjects.

For this reason, we use an extension of the *Subject* field of an ACL entry as introduced in (Biskup & Wortmann 2003). The extension allows the use of algebraic expressions built from principals and local names and the operators *addition* ($+$), *conjunction* ($\&$) and *subtraction* ($-$). It is based on the set-theoretic semantics for SDSI given by (Clarke *et al.* 2001), in which every local name is evaluated to a set of principals. The algebraic operators of the aforementioned extension are interpreted as set-theoretic operations applied to sets of principals. Although the semantics of the *addition* and *conjunction* operators can be implemented in standard SPKI/SDSI(Clarke *et al.* 2001), we claim that using the policy algebra expressions is particularly suitable for specifying and computing with complex access control policies.

The magazine company *EcoMag* allows its subscribers who have a Visa or Diners credit card to buy a special release of the magazine. To do so, *EcoMag* defines an ACL consisting of the following ACL entry:

$<$`Self`,*subject*,`buyspecialrelease`,.,.$>$, with

$$\textit{subject} \;=\; K_{\texttt{EcoMag}}\,\texttt{subscriber} \;\&\; (K_{\texttt{visa}}\,\texttt{card} + K_{\texttt{diner's}}\,\texttt{card})$$

Note, that the subject is an algebraic expression containing the operators *addition* and *conjunction*.

2.4 Integration of DAML-S and SPKI/SDSI

In our approach, Web service providers specify the access control policies of their Web services by using ACLs. In this section we show how ACLs and SPKI/SDSI certificates can be integrated with DAML-S.

Modeling Access Control Lists An ACL is a list of ACL entries. Each ACL entry has the properties `keyholder`, `subject`, `authorization`, `delegation` and `validity`. Refer to class `ACLEntry` in figure 1 and to its specification in OWL in figure 2.

Access Control Policy as Precondition The access control policy of a Web service is a condition that a user must fulfill in order to gain access to the Web service. Further, as identified in section 1.2, access control and functional aspects of a Web service are not always separable (**R3**). That is why we model the access control policy of a Web service as a precondition. In DAML-S, a Web service may have many preconditions, each of which is a sub-property of the property `precondition` of a process. The property `precondition` has range `Condition`, which is currently a place-holder. To model access control policy as precondition, we introduce a class `AccessControlCondition` as subclass of the class `Condition` of DAML-S process model. Other preconditions can be knowledge preconditions or those imposed on functionality related input parameters (Burstein *et al.* 2003; Ankolekar *et al.* 2002). The class `AccessControlCondition` has two important properties, (i) L, an access control list and (ii) I, that refers to the name of the input parameter that is used to carry credentials from a user to a Web service provider. The pair (L, I) is interpreted as: *the credentials in I fulfill the condition specified in the $Subject$ fields of the ACL entries of L.* Refer to class `AccessControlCondition` and its properties `accessControlList` and `inputParameter` in figure 1. By modeling `AccessControlCondition` as subclass of class `Condition` of DAML-S process, we provide a Web service provider a mechanism to specify access control policies in a similar manner as the preconditions related to the functional parameters.

Set of Credentials as Input A user possesses a set of SPKI/SDSI certificates. Using the ACL given in the preconditions of a Web service, she calculates a set of certificates with the help of the chain discovery algorithm as described earlier in section 2.2. A user must send the calculated set (chain) of certificates to a Web service provider in order to prove her eligibility. Note, that unlike the number of functionality related input parameters, the number of certificates a user may send may vary from user to user. Therefore, a Web service provider can not know at the time of describing a Web service, how many certificates a user will send. Therefore, we model a class `SetOfCertificates` with a property `certificate` of multiple cardinality and having the range `SPKICertificate` to carry a set of SPKI/SDSI certificates. Further, we specify classes `SPKINameCertificate` and `SPKIAuthorizationCertificate`

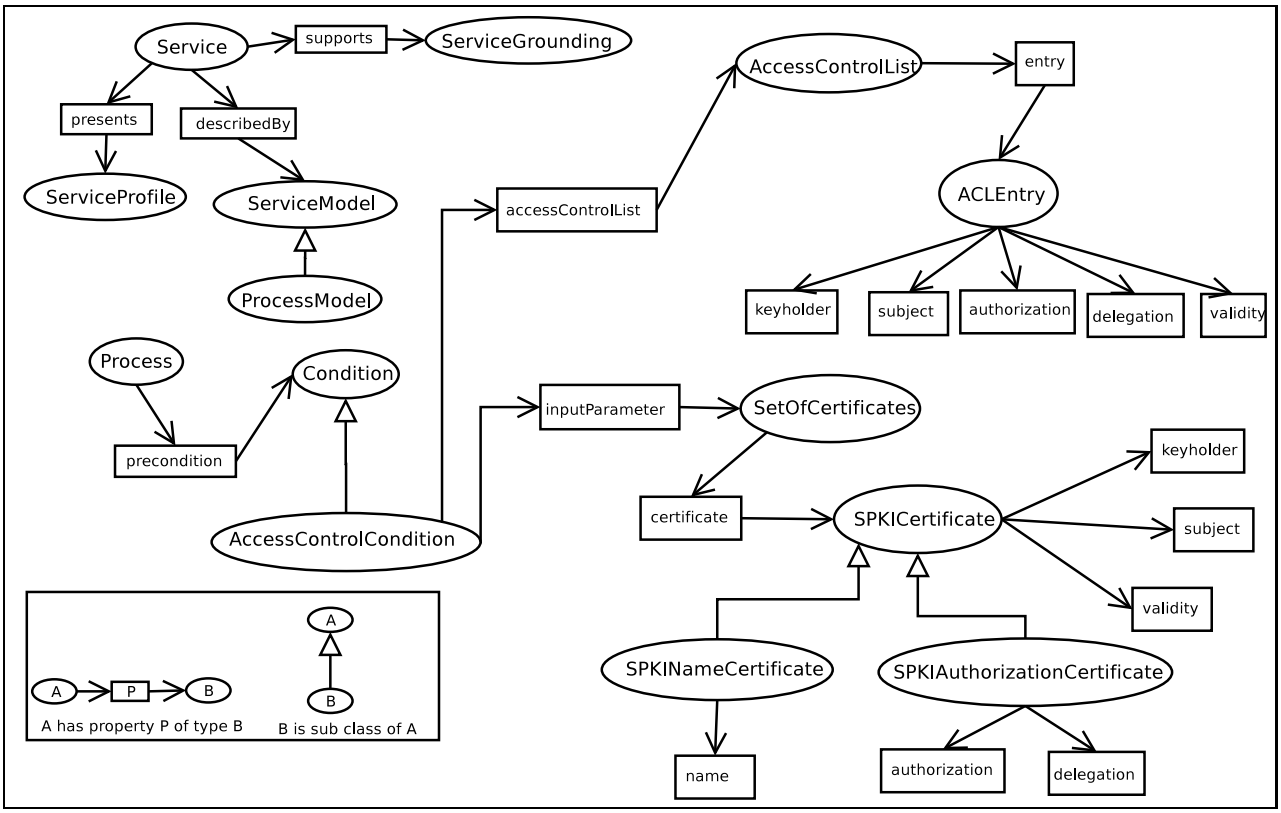

Figure 1: Integration of DAML-S and SPKI

as subclasses of `SPKICertificate`. Refer to classes `SetOfCertificates`, `SPKICertificate`, `SPKINameCertificate` and `SPKIAuthorizationCertificate` in figure 1. A Web service provider will use the class `SetOfSPKICertificates` to specify the input parameter that will be used by the user to send any SPKI certificates.

Output A Web service can return new credentials in addition to the functionality related outputs. To specify such output parameters, a Web service provider can use class `SPKICertificate`. Refer to figure 1. The credentials that are delivered by a component Web service can be used by other component Web services in the same process.

3 Application Scenario

In this section we describe a typical use case of access control of Web services and show how it can be realized with our approach.

3.1 Scenario Description

The magazine company *EcoMag* offers full text access to the electronic edition of the magazine to its subscribers and to customers of a green electricity company. We will show later how we specify this access control policy as an ACL.

The magazine company recognizes its subscribers by a particular membership card issued by the company itself. We model such a membership card as an SPKI/SDSI name certificate.

However, recognizing customers of a green electricity company is not so straight forward assuming that nowadays and in future not only big energy companies but practically everybody can generate and feed electricity into the electricity network. This means that practically everybody can act as a small electricity company spontaneously and dynamically. The problem for the magazine is, how to decide whether a particular electricity company, say *ElecComp*, actually sells green electricity or not. To solve this problem, the magazine trusts a number of organizations, e.g., *EnergyVision* and *GreenPower*, that directly or indirectly issue eco-labels to companies that sell green electricity.

In our application scenario, we model the eco-labels as well as the contract between the electricity company and the customer as SPKI/SDSI name certificates.

If a user, say *Alice*, wants to gain full text access to the magazine's electronic edition she has two options: either she shows her membership card to the web service provider or she needs to prove, that she is a customer of an electricity company, e.g. *ElecComp*, that possesses an eco-label, say either *ok-power* or *gold*, issued by one of the organizations trusted by the magazine company. In order to prove her eligibility, *Alice* shows an appropriate chain of SPKI/SDSI certificates. She calculates the required chain from the Web service's ACL and her set of SPKI/SDSI certificates. She can use the chain discovery algorithm described in section

```
<owl:Class rdf:ID="ACLEntry"/>
<rdfs:subClassOf rdf:resource="&owl;#Thing"/>

<owl:DatatypeProperty rdf:ID="keyholder">
<owl:domain rdf:resource="#ACLEntry"/>
<owl:range rdf:resource="&xsd;#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="subject">
<owl:domain rdf:resource="#ACLEntry"/>
<owl:range rdf:resource="&xsd;#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="authorization">
<owl:domain rdf:resource="#ACLEntry"/>
<owl:range rdf:resource="&xsd;#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="delegation">
<owl:domain rdf:resource="#ACLEntry"/>
<owl:range rdf:resource="&xsd;#string"/>
</owl:DatatypeProperty>

<owl:DatatypeProperty rdf:ID="validity">
<owl:domain rdf:resource="#ACLEntry"/>
<owl:range rdf:resource="&xsd;#string""/>
</owl:DatatypeProperty>

</owl:Class>
```

Figure 2: OWL ontology of Access Control List Entry

2.2 for the calculation by interpreting the web service's ACL entries as SPKI/SDSI authorization certificates.

3.2 Realization in our Approach

The magazine company *EcoMag* specifies its access control policy by an ACL containing the ACL-entry given in figure 3, which is equivalent to the following tuple in the typical SPKI/SDSI notation:

$<$`Self`,*subject*,`fulltextaccess`,`false`,`now`$>$, with

$$\begin{aligned} \textit{subject} \quad = \quad & K_{\texttt{EnergyVision}}\ \texttt{ok-power customer} \\ & + K_{\texttt{GreenPower}}\ \texttt{gold customer} \\ & + \texttt{Self subscriber} \end{aligned}$$

If *Alice* is a subscriber of the magazine *EcoMag*, she could gain full text access by showing her membership card that is specified by the SPKI/SDSI name certificate

$<K_{\texttt{EcoMag}}$,`subscriber`,$K_{\texttt{Alice}}$,`till 2004_12_31`$>$.

If she is a customer of the electricity company *ElecComp* that has received the eco label 'ok-power', she could also gain full text access by showing the following chain (set) of SPKI/SDSI name certificates:

$$\{ \quad <K_{\texttt{EnergyVision}},\texttt{ok-power},K_{\texttt{ElecComp}},.> \ , \\ <K_{\texttt{ElecComp}},\texttt{customer},K_{\texttt{Alice}},.> \quad \}$$

4 Conclusion

We have shown an approach for specifying and using access control with Semantic Web services. Our approach allows

```
<rdf:RDF . . .>
<acp:AccessControlEntry rdf:ID="ace1">
<acp:keyholder>Self</acp:keyholder>
<acp:subject>
     K(EnergyVision) ok-power customer
   + K(GreenPower) gold customer
   + Self subscriber
</acp:subject>
<acp:authorization>fulltextaccess</acp:authorization>
<acp:delegation>false</acp:delegation>
<acp:validity>now</acp:validity>
</acp:AccessControlEntry>
</rdf:RDF>
```

Figure 3: Access control list entry specified in OWL for the full text access web service provided by the magazine company *EcoMag*.

the specification of access control related and functionality related aspects in one unified framework. We classify our work as one of the few contributions that fit in the proof and trust layer of the Semantic Web layer cake. We motivated that authorization based access control is better suited for the Semantic Web than authentication based access control. We identified the requirements for an access control framework for Semantic Web Services. We showed how credentials can be used in the Semantic Web and how SPKI/SDSI can be used to specify credentials. We introduced policy algebra as a mean to specify and handle complex access control policies more efficiently. We showed how SPKI/SDSI credentials be modeled and integrated with DAML-S. We gave an application scenario and demonstrated how our approach works within a concrete setting.

Although our approach is technically mature enough to be used in scenarios with simple Web services, there are still some issues that need to be investigated in more detail to enable more complex applications. For example, in case of a composite Web services, there must be some (semi-) automatic mechanism to calculate the access control policy of a composite Web service from its control flow graph, data flow graph and the access control policies of its component Web services. We see the problem of dealing with credentials that are delivered as output or consumed by a component Web service of a composite Web service as another important problem that should be investigated in the future.

We did not deal with discovery of Web services in this paper but believe that existing approaches can still be used when slightly modified for example as follows. A discovery algorithm can assume all access control related conditions to be true and calculate a set of Web services that offer the required functionality. The discovering component can then send the preconditions of the relevant Web services to the user, who can check the satisfiability of the preconditions on the basis of her set of credentials.

Our long-term goal is to investigate the compositionality of semantic Web services and access control policies. We believe, that specification of access control policies and Semantic Web services in one unified framework is a necessary requirement for further research in the aforementioned area.

Acknowledgements

We thank Joachim Biskup, Thomas Leineweber, Ralf Menzel, Lars Schmidt-Thieme and Rudi Studer for helpful discussions and proof reading. A part of this work was funded by the SESAM Internetökonomie project of the Federal Ministry of Education and Research (BMBF).

References

Ankolekar, A.; Burstein, M. H.; Hobbs, J. R.; Lassila, O.; Martin, D.; McDermott, D. V.; McIlraith, S. A.; Narayanan, S.; Paolucci, M.; Payne, T. R.; and Sycara, K. 2002. DAML-S: Web Service Description for the Semantic Web. In *ISWC2002: Ist International Semantic Web Conference, Sardinia, Italy*, Lecture Notes in Computer Science, 348–363. Springer.

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The semantic web. *Scientific American*.

Bishop, M. 2003. *Computer Security – Art and Science*. Addison Wesley.

Biskup, J., and Karabulut, Y. 2002. A hybrid PKI model with an application for secure mediation. In *Proc. of the 16th Annual IFIP WG 11.3 Working Conference on Data and Application Security*. King's College, Cambridge, UK.

Biskup, J., and Wortmann, S. 2003. Towards a credential-based implementation of compound access control policies. Technical report, University of Dortmund. http://ls6-www.cs.uni-dortmund.de/issi/publications.

Blaze, M.; Feigenbaum, J.; Ioannidis, J.; and Keromytis, A. D. 1999. The role of trust management in distributed systems security. In *Secure Internet Programming: Issues in Distributed and Mobile Object Systems*, volume 1603 of *Lecture Notes in Computer Science*, 183–210. Springer Verlag, Berlin.

Brands, S. 2000. *Rethinking Public Key Infrastructures and Digital Certificates*. MIT Press, Cambridge-London.

Burstein, M.; Denker, G.; Hobbs, J.; Kagal, L.; Lassila, O.; Martin, D.; McDermott, D.; McIlraith, S.; Narayanan, S.; Paolucci, M.; Parsia, B.; Payne, T.; Sirin, E.; Srinivasan, N.; and Sycara, K. 2003. Daml-s: Semantic markup for web services, version 0.9. Technical report, DAML-S Services Coalition.

Chaum, D. 1985. Security without identification: transaction systems to make big brother obsolete. In *Communications of the ACM 28*, volume 10, 1030–1044. ACM Press.

Clarke, D. E.; Elien, J.-E.; Ellison, C. M.; Fredette, M.; Morcos, A.; and Rivest, R. L. 2001. Certificate chain discovery in SPKI/SDSI. *Journal of Computer Security* 9:285–322.

Denker, G.; Kagal, L.; Finin, T.; Sycara, K.; and Paoucci, M. 2003. Security for daml web services: Annotation and matchmaking. In *Second International Semantic Web Conference*, volume 2870 of *Lecture Notes in Computer Science*. Springer.

Denning, D. E. 1982. *Cryptography and Data Security*. Addison Wesley.

Ellison, C. M.; Frantz, B.; Lampson, B.; Rivest, R. L.; Thomas, B. M.; and Ylonen, T. 1999a. Simple public key certificate. http://world.std.com/cme/html/spki.html.

Ellison, C. M.; Frantz, B.; Lampson, B.; Rivest, R. L.; Thomas, B. M.; and Ylonen, T. 1999b. SPKI certificate theory. Internet RFC 2693.

Fensel, D.; Hendler, J.; Lieberman, H.; and Wahlster, W., eds. 2002. *Spinning the Semantic Web*. MIT Press.

Kagal, L.; Finin, T.; and Joshi, A. 2003. A policy based approach to security for the semantic web. In *2nd International Semantic Web Conference (ISWC2003)*, volume 2870 of *Lecture Notes in Computer Science*. Springer.

Patel-Schneider, P., and Fensel, D. 2002. Layering the semantic web: Problems and directions. In *ISWC2002: Ist International Semantic Web Conference, Sardinia, Italy*, Lecture Notes in Computer Science, 16–29. Springer.

Rivest, R. L., and Lampson, B. 1996. SDSI - a simple distributed security infrastructure. http://theory.lcs.mit.edu/cis/sdsi.html.

Samarati, P., and Capitani di Vimercati, S. 2001. Access control: policies, models, and mechanisms. In Focardi, R., and Gorrieri, R., eds., *Foundations of Security Analysis and Design (FOSAD)*, volume 2171 of *Lecture Notes in Computer Science*, 137–196. FOSAD 2000, Bertinoro, Italy.

Samarati, P. 2002. Enriching access control to support credential-based specifications. In Sigrid Schubert, B. R., and Jesse, N., eds., *"Informatik bewegt" Proc. of the 32. Jahrestagung der Gesellschaft für Informatik*, volume P-19 of *Lecture Notes in Informatics*, 114–119. German Informatics Society (GI), Dortmund, Germany. http://ls6-www.cs.uni-dortmund.de/issi/cred_ws/index.html.de.