# On Specifying Security Policies for Web Documents with an XML-based Language*


Elisa Bertino
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico, 39/41
20135 Milano Italy
bertino@dsi.unimi.it

Silvana Castano
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico, 39/41
20135 Milano Italy
castano@dsi.unimi.it

Elena Ferrari
Dipartimento di Scienze dell'Informazione
Università degli Studi di Milano
Via Comelico, 39/41
20135 Milano Italy
ferrarie@dsi.unimi.it



## ABSTRACT

The rapid growth of the Web and the ease with which data can be accessed facilitate the distribution and sharing of information. Information dissemination often takes the form of documents that are made available at Web servers, or that are actively broadcasted by Web servers to interested clients. In this paper, we present an XML-compliant formalism for specifying security-related information for Web document protection. In particular, we introduce $\mathcal{X}$-Sec, an XML-based language for specifying subject credentials and security policies and for organizing them into subject profiles and policy bases, respectively. The language is complemented by a set of subscription-based schemes for accessing distributed Web documents, which rely on defined XML subject profiles and XML policy bases.



## Keywords

XML, Access control, Subject credentials, Security policies.


## 1. INTRODUCTION

Companies and organizations are today starting to use the Web as the main information dissemination means both at internal and external level. Information dissemination often takes the form of documents that are made available at Web servers, or that are actively broadcasted by Web servers to interested clients. Furthermore, documents may be exchanged among the various servers. Because, however, documents often contain information at different degrees of sensitivity, there is a strong need for models and mechanisms enabling the specification and enforcement of access control policies. Protection of documents in such an environment entails addressing several requirements. In particular, main protection requirements for Web documents that influence the definition of the policies for their access and exchange are related to the following characteristics.


*This work has been partially supported by a grant from Microsoft Research.




*SACMAT'01,* May 3-4, 2001, Chantilly, Virginia, USA.



**Protecting document contents in a selective and differentiated way.** Web documents may have a nested or hierarchical structure, being defined in terms of components that can be themselves organized into subcomponents. Moreover, Web documents are inter-linked, to allow hypertextual navigations across related documents. For example, an XML [16] document is defined in terms of elements and attributes; document elements can contain elements (subelements) in turn originating a hierarchical structure. An XML document can be linked to other related documents or elements through `IDREF(S)` attributes or Xlinks. As another example, HTML pages of a Web site are inter-related through hypertextual links.
Very often different components of the same Web document have varying protection requirements. For example, a Web document describing a purchase order can provide descriptive information about the order and about the item(s) associated with it; information about the name of (all or part of) items could be made available to everyone, whereas information regarding the carrier should be released only to selected subjects. The purchase order document can be linked to the documentation of its related clients; such link could be kept hidden from most subjects and made visible only to a restricted number of authorized subjects. To support a differentiated and selective protection of Web documents, security policies must apply to fine-grained protection objects, identified on the basis of both the structure of the document and the document content. Examples of protection objects in case of XML documents are the whole document, a set of documents, an element of a document, an attribute of a document. Protection objects of an HTML document can be the whole site, a single page, a section within a page, a link between pages. Moreover, to reduce the number of policies to be defined, security policies must support a notion of *propagation.* Policies specified for a protection object at a given granularity level propagate by default to all protection objects related to it according to a certain relationship in the structure. For example, a security policy specified for an element of an XML document applies by default to all its subelements.

**Protecting documents at the intensional level.** Formalisms adopted for the representation of Web documents can allow an explicit description of the document structure at the intensional level. For example, in the case of XML documents, an XML DTD or an XMLSchema can be associated with a set of valid XML documents. An intensional description of the structure of HTML pages of Web sources can also be provided using suitable data models allowing to describe the logical organizations of data in Web pages. Structure descriptions associated with a Web source can be exploited to concisely define security policies at the intensional level, that hold for all Web documents conforming to this intensional description. For example, it is possible to specify security policies at the DTD level, which apply to all valid documents conforming to the considered DTD. In this way, the definition of policies for Web documents exploits a notion of schema or type, in analogy with conventional policies for relational and object-oriented databases [5].

**Specifying subjects by means of credentials.** The population accessing Web document sources is generally composed of heterogeneous subjects, characterized by different skills and needs. Moreover, the population is dynamic, in that the number and type of subjects is not known a priori and can change over time very frequently. In this context, conventional identity-based access control schemes are not sufficient and security policies based on the notion of *credential* are required [14]. Subject credentials assert arbitrary properties of a subject, either personal characteristics, or characteristics and properties deriving from relationships the subject has with other subjects (e.g., qualifications within an organization). In this way, the specification of security policies becomes more direct and intuitive, since they are defined in general terms, close to the rules and conventions holding for the documents to be protected.

**Supporting multiple schemes for distributed access.** Web documents can be stored at one source or can be distributed across several sources over the Internet. Different distributed access schemes are thus necessary to enforce access control based on subject credentials. For example, Web document sources can be accessed according to subscription-based schemes, such as in the case of digital library systems. In such a scheme, subjects are required to register with a source of interest, during a mandatory subscription phase, for the definition of their subject credentials. As another example, according to a negotiation-based scheme, a requesting subject can access documents in a source by presenting its credentials directly. A source can be locally responsible for accepting credentials presented by a subject (decentralized approach) or can refer to a third party for the appropriate certificate (centralized approach). Using subject credentials for access control to Web documents requires to define who is responsible for credential issuing (e.g., a third party), how properties asserted by the credentials can be certified [11], and other issues related to trust management in distributed systems [4, 14].

The goal of the paper is thus to present $\mathcal{X}$-Sec, a language based on XML for specifying subject credentials and security policies for Web documents, to fulfill the requirements outlined above. Moreover, the paper describes subscription-based schemes for accessing distributed Web documents, based on XML credentials and security policies .

The work presented in this paper has been developed in the framework of the Author-$\mathcal{X}$ project [2]. Author-$\mathcal{X}$ is a Java-based system for access control and security policy design for XML documents. For access control, Author-$\mathcal{X}$ supports policy specification at varying granularity levels, according to the model presented in [1]. Additionally, Author-$\mathcal{X}$ supports the push and pull distribution policies for document release. Another relevant feature of Author-$\mathcal{X}$ is the support for distributed document updates through the combination of hash functions and digital signature techniques. A number of administration tools are also provided, to facilitate security administration according to the underlying security policies. What described in this paper is the extension of Author-$\mathcal{X}$ for enforcing access control based on subject credentials.

The remainder of the paper is organized as follows. Section 2 presents the XML-compliant formalism for specifying subject credentials. Section 3 presents the XML-compliant formalism for specifying access control policies, and gives an example of its application for the protection of XML document sources. Section 4 presents two subscription schemes according to which Web documents can be accessed and retrieved, based on XML subject credentials and XML security policies. Section 5 surveys related work. Finally, Section 6 concludes the paper and outlines future work directions.

## 2. $\mathcal{X}$-Sec CREDENTIAL SPECIFICATION

In this section we present the $\mathcal{X}$-Sec language and related formalism for the specification of subject credentials in XML. We first introduce the concept of *credential-type*, as a way to simplify the task of credential specification, then we present the XML representation of credentials and credential-types. In particular, a credential-type is modeled as a DTD and a specific credential as a valid XML document conforming to the DTD representing the corresponding credential-type. Finally, we introduce $\mathcal{X}$-profiles, useful for the evaluation of credential-based security policies.

### 2.1 $\mathcal{X}$-Sec credential-types

Credentials with a similar structure are grouped into credential-types. Examples of credential-types are `business manager`, `customer`, and `carrier`. To formalize credential-types, let $CN$ be a set of names of credential-types and $PN = SP \cup CP$ be a set of property names of credential-types, where $SP$ denotes a set of simple property names and $CP$ a set of composite property names, respectively. A credential-type can be conceptualized as a pair $\langle n_{ct}, P_{ct}\rangle$, where $n_{ct}$ is the name of the credential-type $ct$ and $P_{ct}$ is a set of property specifications for $ct$. A property specification provides the name and the domain of a property. Credential-type properties can be either simple or composite. Simple properties take values from basic domains (e.g.,`integer`, `string`) whereas composite properties take value from domains defined by applying conventional constructors (e.g., `set, record, list`) on basic domains.

Credential-types in our XML-compliant formalism correspond to DTDs and are formally defined as graphs as follows. Let *Label* be a set of element tags and attribute names, and $Label^*$ be the set of strings obtained through the concatenation of names in *Label* and a symbol in $\{*, +, ?\}$.

DEFINITION 1. (XML Credential-type). *Given a credential-type $ct = \langle n_{ct}, P_{ct}\rangle$, an XML credential-type $\mathcal{X}$-ct is a tuple $(\bar{v_{ct}}, V_{ct}, E_{ct}, \phi_{E_{ct}})$, where:*

```
<!DOCTYPE carrier_employee[
<!ELEMENT carrier_employee (name,address,
          phone_number*,email?,company)>
<!ELEMENT name (fname,lname)>
<!ELEMENT address (#PCDATA)>
<!ELEMENT phone_number (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT company (#PCDATA)>
<!ELEMENT fname (#PCDATA)>
<!ELEMENT lname (#PCDATA)>
<!ATTLIST carrier_employee credID
                     ID #REQUIRED>
<!ATTLIST carrier_employee CIssuer
                     CDATA #REQUIRED >
]>
```

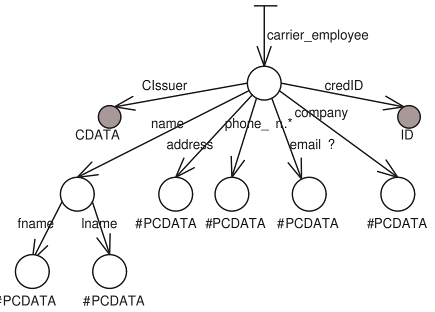


**Figure 1: (a) Example of XML credential-type and (b) its corresponding graph representation**

- $\bar{v_{ct}}$ *is the root element denoting the whole ct.* $\bar{v}_t$ *has two default attributes, namely the* `credID` *attribute of type* `ID`*, specifying credential identifiers, and the* `CIssuer` *attribute of type* `CDATA`*, identifying the issuer of credentials of type ct;*

- $V_{ct} = V_{ct}^e \cup V_{ct}^a$ *is a set of nodes representing elements and attribute types of ct. In particular, each node in* $V_{ct}^e$ *which is a direct child of* $\bar{v_{ct}}$ *corresponds to a property of ct: if the child is a leaf node, then it represents a simple property of ct; if it is an intermediate node, then it corresponds to a composite property. The number of direct children of* $\bar{v_{ct}}$ *in* $V_{ct}^e$ *is equal to the number of properties (both simple and composite) of ct.*

- $E_{ct} \subseteq V_{ct} \times V_{ct}$ *is a set of edges, where* $e \in E_{ct}$ *represents an element-subelement or an element-attribute relationship;*

- $\phi_{E_{ct}} : E_{ct} \rightarrow Label^*$ *is the edge labeling function. In particular: the edge entering* $\bar{v_{ct}}$ *is labeled with* $n_{ct}$*; edges entering leaf nodes in* $V_{ct}^e$ *that are direct children of* $\bar{v_{ct}}$ *are labeled with names in SP; edges entering intermediate nodes in* $V_{ct}^e$ *that are direct children of* $\bar{v_{ct}}$ *are labeled with names in CP.* □

An XML credential-type is thus a DTD where simple properties are modeled as empty elements and composite properties as elements with element content, whose subelements model composite property components. Figure 1(a) shows the XML credential-type `carrier_employee`. This credential-type contains the basic information about a carrier employee, namely, his/her name, working address, phone number, email, and company name. With reference to Figure 1(a), `<!ELEMENT email #PCDATA>` is an example of simple property, representing the email of a carrier employee, whereas `<!ELEMENT name (fname, lastname)>` is an example of composite property, because name is composed by `fname` and `lname` properties. Figure 1(b) gives the graph representation of the XML credential-type in Figure 1(a). In the graph representation, elements are represented as white circles whereas attribute types are represented as gray circles. Edge labels followed by "*" represent repeatable elements (e.g., `phone_number`), whereas edge labels followed by "?" represents optional elements (e.g., `email`).

## 2.2 $\mathcal{X}$-Sec credentials and $\mathcal{X}$-profiles

A credential is an instance of a credential-type, and specifies the set of property values characterizing a given subject against the credential-type itself. Credentials are certified by the credential issuer (e.g., a certification authority) using standard digital signature techniques [12]. The credential issuer is also responsible for certifying properties asserted by credentials themselves. In the XML-based formalism, a credential is an XML document containing values for all the subject properties (simple or composite) specified in the corresponding credential-type. Consequently, a credential can be thought of as an instance of an XML credential-type, as formalized by the following definition.

DEFINITION 2. (XML credential). *Given an XML credential-type* $\mathcal{X}$*-ct, an XML credential* $\mathcal{X}$*-cred of type ct is a valid document wrt* $\mathcal{X}$*-ct, represented as a tuple* $(\bar{v}_c, V_c, E_c, \phi_{E_c})$*, where:*

- $\bar{v}_c$ *is the root element denoting the whole credential. The* `credID` *attribute value of* $\bar{v}_c$ *denotes the credential identifier, whereas the* `CIssuer` *attribute value of* $\bar{v}_c$ *is the digital signature of the issuer of* $\mathcal{X}$*-cred;*

- $V_c = V_c^e \cup V_c^a$ *is a set of nodes representing elements and attributes, respectively. Each* $v \in V_c^a$ *has an associated value conforming to the type of the corresponding attribute node in* $\mathcal{X}$*-cred;*

- $E_c = E_c^e \cup E_c^a \subseteq V_c \times V_c$ *is a set of edges, where* $e \in E_c^e$ *is an edge representing an element-subelement relationship whereas* $e \in E_c^a$ *is an edge representing an element-attribute relationship;*

- $\phi_{E_c} : E_c \rightarrow Label$ *is the edge labeling function, operating as the labeling function defined in* $\mathcal{X}$*-ct.* □

Figure 2(a) shows an example of XML credential, instance of the XML credential-type in Figure 1, whereas Figure 2(b) presents its graph representation.

To simplify the process of evaluating subject credentials against security policies, all the credentials a subject possesses are collected into the so called $\mathcal{X}$*-profile*, formally defined as follows.

DEFINITION 3. ($\mathcal{X}$-profile). *Let s be a subject, and let* $\mathcal{C}(s) = \{\mathcal{X}\text{-}cred_1, \ldots, \mathcal{X}\text{-}cred_n\}$ *be a collection of XML credentials associated with s, where* $\mathcal{X}\text{-}cred_i = (\bar{v}_{ci}, V_{ci}, E_{ci}, \phi_{E_{ci}})$,

```
<carrier_employee credID=‘‘154", CIssuer = ‘‘CA16">
   <name>
      <fname> Bob </fname>
      <lname > Watson </lname>
   </name>
   <address> 24 Baker Street </address>
   <phone_number> 8005769840 </phone_number>
   <email> bwatson@ups.com </email>
   <company> UPS </company>
</carrier_employee>
```

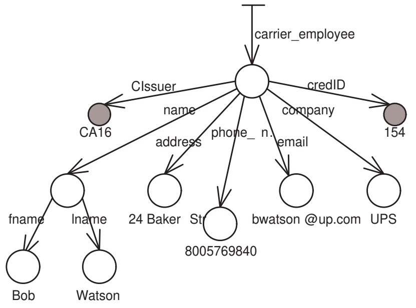


**Figure 2: (a) Example of XML credential and (b) its corresponding graph representation**

```
<X-profile sbjID=‘‘bw585", PIssuer = ‘‘CA16">
   <carrier_employee credID=‘‘154", CIssuer = ‘‘CA16">
      <name>
         <fname> Bob </fname>
         <lname > Watson </lname>
      </name>
      <address> 24 Baker Street </address>
      <phone_number> 8005769840 </phone_number>
      <email> bwatson@ups.com </email>
      <company> UPS </company>
   </carrier_employee>
   <stockholder credID=‘‘254", CIssuer = ‘‘CA16">
      <name> ... </name>
      <company> Paragon </company>
         <stocknumber> 400 </stocknumber>
         <stockvalue> $900 </stockvalue>
      <company> ... </company>
   </stockholder>
</X-profile>
```

**Figure 3: An example of $\mathcal{X}$-profile**

*for $i = 1, \ldots, n$. The $\mathcal{X}$*-profile *of $s$ is a well-formed XML document $(\bar{v}_p, V_p, E_p, \phi_{E_p})$, where:*

- *$\bar{v}_p$ is the root element denoting the whole $\mathcal{X}$*-profile*. The root element has two attributes, namely* `sbjID` *and* `PIssuer`*. The* `sbjID` *attribute value of $\bar{v}_p$ denotes the subject identifier of $s$, whereas the* `PIssuer` *attribute value of $\bar{v}_p$ is a string denoting the issuer authority of the $\mathcal{X}$*-profile*;*[1]
- *$V_p = \bigcup_{i=1}^{n}\{\bar{v}_{ci}\} \cup \bigcup_{i=1}^{n} V_{ci}$ is a set of nodes representing elements and attributes, respectively, where each direct child of $\bar{v}_p$ is the root element of an XML credential in $\mathcal{C}(s)$;*
- *$E_p = E_p^e \cup E_p^a \subseteq V_p \times V_p$ is a set of edges, where $e \in E_p^e$ is an edge representing an element-subelement relationship, whereas $e \in E_p^a$ is an edge representing an element-attribute relationship;*
- *$\phi_{E_p} : E_p \rightarrow Label$ is the edge labeling function. In particular: the edge entering $\bar{v}_p$ is labeled with* `X-profile`*, and the edges entering intermediate nodes that are direct children of $\bar{v}_p$ are labeled with $\mathcal{X}$-cred$_1, \ldots, \mathcal{X}$-cred$_n$.* □

[1]Similar to credentials, we suppose that $\mathcal{X}$-profiles are issued by a certification authority. In fact, in open systems, many different authorities could release credentials to the same subject. In this case, we suppose that an authority is responsible for the certification of the whole $\mathcal{X}$-profile.

EXAMPLE 1. *With reference to the example in Figure 2, suppose that subject* `Bob Watson`*, besides being an employee of the UPS company, is also a stockholder of a company, named* `Paragon`*, which is a client of UPS. Figure 3 shows his $\mathcal{X}$-profile.* ○

# 3. $\mathcal{X}$-Sec SECURITY POLICY SPECIFICATION

In this section, we first present an XML template for specifying credential-based security policies for web documents, based on the protection requirements outlined in the introduction. The template has been conceived to be as general as possible to be able to model security policies for a variety of web documents (e.g., HTML, XML documents). Then, as an example of the template applicability, we show how such template can be instantiated for the protection of XML documents.

## 3.1 $\mathcal{X}$-Sec policy base template

Security policies specify which subjects can exercise which privileges on which documents within a document source.

| Name | Type | Value |
|---|---|---|
| `cred_expr` | `CDATA` | Xpath expression on $\mathcal{X}$-profiles |
| `target` | `CDATA` | denotes the document(s) to which the policy applies |
| `path` | `CDATA` | denotes selected portions within the `target` document(s) |
| `priv` | `CDATA` | specifies the security policy access mode |
| `type` | `CDATA` | specifies whether the security policy is positive or negative |
| `prop` | `CDATA` | specifies the propagation option of the security policy |

**Table 1: Attribute specification**

```
<!DOCTYPE policy_base[
<!ELEMENT policy_spec EMPTY>
<!ATTLIST policy_spec cred_expr CDATA #REQUIRED>
<!ATTLIST policy_spec target CDATA #REQUIRED>
<!ATTLIST policy_spec path CDATA>
<!ATTLIST policy_spec priv CDATA #REQUIRED
<!ATTLIST policy_spec type CDATA #REQUIRED
<!ATTLIST policy_spec prop CDATA #REQUIRED
]>
```

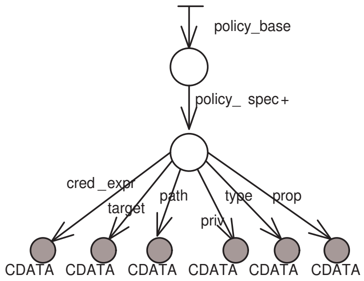


**Figure 4: (a) XML policy base template and (b) its corresponding graph representation**

To keep our template as general as possible, we suppose that policies can be of two different kinds: *positive policies*, to specify authorizations, and *negative policies*, to specify denials. Moreover, since web documents may have a nested or hierarchical structure, the policy base template supports the specification of *propagation options* for the security policies, that specify how a policy specified for a given document portion propagates to other document portions related to it according to a certain relationship in the document. This can be a useful feature for reducing the number of policies that need to be specified.

Based on the above considerations an XML policy base template can be formally defined as follows.

DEFINITION 4. (XML policy base template). *An XML policy base template $\mathcal{X}$-pt is a tuple $(\bar{v_{pt}}, V_{pt}, E_{pt}, \phi_{E_{pt}})$, where:*

- *$\bar{v_{pt}}$ is the root element denoting the whole policy base.*
- *$V_{pt} = V^e_{pt} \cup V^a_{pt}$ is a set of nodes representing elements and attribute types. In particular, $\bar{v_{pt}}$ has a unique direct child, belonging to $V^e_{pt}$, which models a security policy specification. Such node has five attributes whose specification is given in Table 1.*
- *$E_{pt} \subseteq V_{pt} \times V_{pt}$ is a set of edges, where $e \in E_{pt}$ represents an element-subelement or an element-attribute relationship;*
- *$\phi_{E_{pt}} : E_{pt} \rightarrow Label^*$ is the edge labeling function. In particular: the edge entering $\bar{v_{pt}}$ is labeled with* `policy_base`*; the edge entering the (unique) direct child of $\bar{v_{pt}}$ is labeled* `policy_spec`$^+$*, to denote the fact that this node is repeatable.* □

An $\mathcal{X}$-Sec policy base template (cfr. Figure 4) is thus a DTD where policy specifications are modeled as empty elements having an attribute for each policy component (i.e., subject, object, privilege, type, and propagation).

A policy base is an instance of the XML policy base template, as formalized by the following definition.

DEFINITION 5. (XML policy base). *Given an XML policy base template pt, an XML policy base is a valid XML document wrt pt.* □

### 3.2 Policy base template instantion for the protection of XML sources

In this section, we show how the XML policy base template can be instantiated for the protection of XML sources. The instantiation is based on the access control model proposed in [1], extended for the support of credential-based security policies. Being the target the protection of XML sources, objects to be protected can be either XML documents or DTDs (or portions of them), where an access control policy specified for a DTD implies an analogous policy for all the valid documents that conform to this DTD. Thus, the value of the `target` attribute of the `policy_spec` element in an XML policy base for the protection of XML sources is either the name of an XML document or DTD, whereas the `path` value is an Xpath expression [15] selecting specific portions within the target document/DTD. We classify security policies for XML sources into two groups: *browsing policies*, that allow one to see the information in a document and/or to navigate through its links, and *authoring policies* that allow the modification of XML documents under different modes. Thus, attribute `priv` takes a value in the following set {`view`,`navigate`,`append`,`write`, `all`}, where the `view` privilege authorizes a subject to view an element and/or (some of) its components, the `navigate` privilege authorizes a subject to see the existence of a specific link or of all the links in a given element and to navigate through them, the `append` privilege allows a subject to write information in an element (or in some of its parts) or to include a link in an element, without deleting any pre-existing information, the `write` privilege allows a subject to modify the content of an element and to include links in the element, and the

```
<policy_base>
   <policy_spec cred_expr="//secretary [department="sales"]" target="Purchase_order.dtd"
      priv="ALL" type="GRANT" prop="CASCADE"/ >
   <policy_spec cred_expr="//carrier_employee[company="UPS"]" target="Purcahse_order.xml"
      path = "//Purchase_order[Purchase_order/carrier/name="UPS"]" priv="VIEW" type="GRANT"
      prop="CASCADE"/ >
   <policy_spec cred_expr="//carrier_employee[company="UPS"]" target="Purchase_order.xml"
      path="//item" priv="VIEW" type="DENY" prop="CASCADE"/ >
   <policy_spec cred_expr="//publicity_agent target="Purchase_order.dtd"
      path="//item/description priv="VIEW" type="GRANT" prop="NO_PROP"/ >
   <policy_spec cred_expr="//publicity_agent target="Purchase_order.dtd"
      path="//Purchase_order/orderID" priv="VIEW" type="GRANT" prop="NO_PROP"/ >
</policy_base>
```

**Figure 5: An example of policy base**

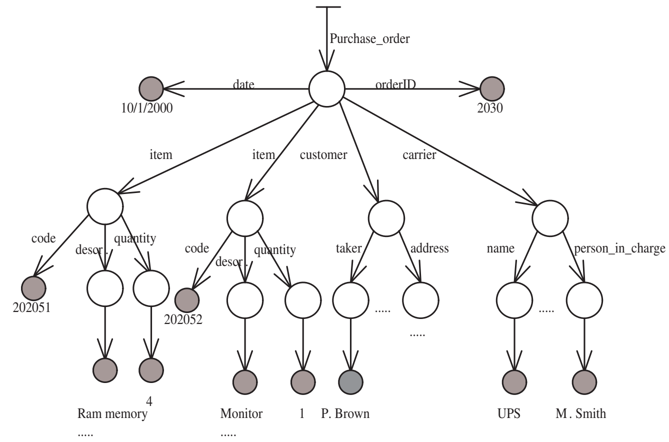


**Figure 6: XML representation of a purchase order**

`all` privilege subsumes all the other privileges. Due to the hierarchical structure of both XML documents and DTDs, propagation options can be specified for a policy that state how a policy defined at a given level in the document/DTD hierarchy propagates to lower level. We consider a spectrum of propagations consisting of three different options, that result into three different values for attribute `prop`: NO_PROP, when the policy only applies to the protection objects which appear in its specification; FIRST_LEV when the security policy propagates to all the direct subelements of the elements in the policy specification, and CASCADE, when the policy propagates to all the direct and indirect subelements of the elements in the policy specification.

An example of $\mathcal{X}$-Sec policy base for the protection of XML sources is given in Figure 5. Such policy base refers to an XML source in the supply domain. More precisely, the security policies refer to the purchase order document whose graph representation is given in Figure 6.

According to the policies in the policy base of Figure 5, the secretaries working in the sales department can modify and access all the purchase order documents (such protection requirement is enforced through a policy specified at the DTD level), UPS employees can access the information about the customer, the carrier, and the order id and date contained in the purchase orders for which UPS is the carrier. Note that, this latter requirement is concisely enforced using two policies: a positive policy which applies to the whole purchase order documents referring to orders for which UPS is the carrier, and a negative policy which applies only to the portions of the purchase order documents that must not be made available to UPS employees (i.e., the item elements). Finally, the last two policies authorize publicity agents to access the order identifier and the description of the items contained in purchase orders, respectively.

# 4. SUBSCRIPTION SCHEMES FOR A CONTROLLED ACCESS TO WEB DOCUMENTS

In this section, we outline two different schemes according to which Web documents of one or more sources can be accessed and retrieved by subjects, based on XML subject credentials and XML security policies. In such schemes, document accesses are subordinated to a mandatory initial subscription of the subjects to a document source of interest, where subject credentials are defined. Then, credential-based access control is performed following different strategies in the proposed schemes. In the following, we first describe the proposed schemes, we then briefly illustrate access control strategies.

## 4.1 The A2O and A2M schemes

The schemes we propose (cfr. Table 2) are conceived to allow a subject to access Web document sources in different ways, to support protection requirements of different application domains (e.g., digital library, e-commerce, Web-based information systems). The first scheme is called *access-to-one (A2O) scheme* and is designed for a centralized environment where the documents to be released reside at a single site. The second scheme is called *access-to-many (A2M) scheme* and is conceived for a federated environment where access requests can be answered by retrieving documents from a federation of different sources.

**The A2O scheme.** The A2O scheme allows a subject to access Web documents stored in a target source by subscribing it. The source is responsible for managing the policy and credential bases for the Web documents to be protected. The source defines its own XML credential-types and generates XML subject credentials based on information supplied by subjects during the subscription phase. XML credentials encoded in the $\mathcal{X}$-Sec credential base and XML security policies encoded in the $\mathcal{X}$-Sec policy base are used for governing access to documents upon subject requests. In the A2O scheme, the expected interactions between a subject $s$ and a target source $T$ are the following (see Table 3): the *subscription*, that takes place only once, during which $s$ supplies the

| Access Scheme | Target | Req.ts for the Access | Retrievable Docs. | Application Domain |
|---|---|---|---|---|
| A2O | One source ($T$) | Subscription to $T$ | Docs. of $T$ | • Digital library<br>• E-commerce<br>• Web-based IS |
| A2M | Federation of sources | Subscription to the master source of the federation | Docs. of all sources | |

**Table 2: Subscription schemes for XML-based access control to Web documents**

| Interactions | Interacting Parties | Initiator | Exchanged Info | Result |
|---|---|---|---|---|
| Subscription | Subject ($s$) and Target source $T$ | $s$ | Info for credentials | • Subscription to $T$<br>• X-profile of $s$ |
| Access control | Subject ($s$) and Target source $T$ | $s$ | Access request ($r$) | • View of the authorized target document(s) in $r$ |

**Table 3: Expected interactions in the A2O scheme**

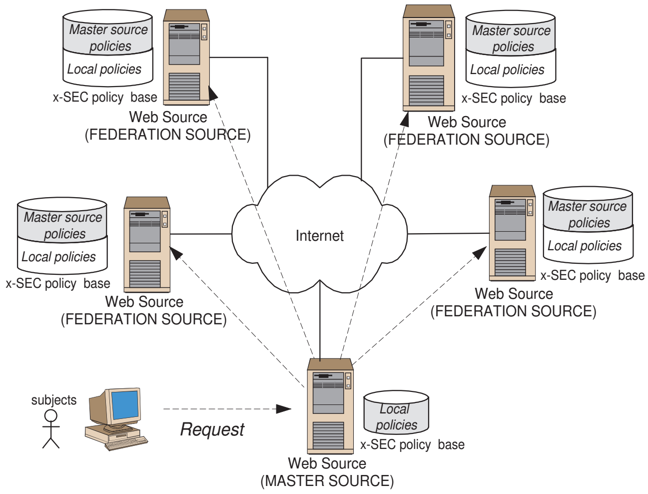


**Figure 7: Access control in the A2M scheme with shallow agreement**

information requested by $T$ to define $s$'s credential(s) to be encoded in the credential base; the *access control*, that takes place each time $s$ submits an access request to $T$ for target document(s).

**A2M scheme.** The A2M scheme is an extension of the previous one, to accomodate the case of Web documents distributed among several sources forming a *federation*. Each source of the federation manages its own $\mathcal{X}$-Sec policy and credential bases and is responsible for their definition. The expected interactions in this scheme are summarized in Table 4. In particular, we distinguish a source ($S_m$) with the role of *master source*, to which all subjects must subscribe (subscription phase). Subjects submit access requests to the master source. The master source processes an access request locally, against its document source. Moreover, it forwards the initial subject requests to all or part of the federation sources (access control phase). Besides the subscription phase and the access control phase, the A2M scheme requires also an agreement phase between the master source and each federation source. The *agreement phase* (performed at the time a source joins the federation) has the goal of reaching an agreement about the security policies a federation source should use for processing an access request received from the master source. We support two modes for performing the agreement. A *shallow agreement* requires the definition of additional security policies, for governing the accesses of the master source to the documents stored at each federation source. In this mode, access requests of different subjects forwarded by the master source to a given federation source are treated uniformly at this latest source, according to the master source policies specified at this source (see Fig. 7). The *deep agreement* requires the definition of mappings between the $\mathcal{X}$-Sec credential-types defined at the master source and the $\mathcal{X}$-Sec credential-types defined at each federation source, to guarantee semantic conversion of, possibly heterogeneous, subject credentials. The goal is to make subject credentials defined at the master source understandable by a given federation source. In this mode, access requests of subjects with different credentials forwarded by the master source to a given federation source are treated differently at this latest source, on the basis of mapped credential-types. Under the deep agreement mode, the $\mathcal{X}$-Sec credential base of the master source needs to be extended with *conversion rules*, to enforce required credential-type mappings.

EXAMPLE 2. *Suppose to consider the credential-type of Fig. 2 describing a carrier employee role and an additional credential-type shown in Fig. 8, describing a generic employee role. The purpose of the deep agreement phase is to define mappings to convert a credential-type of the master source (e.g.,* `employee`*) into another one of a federation source (e.g.,* `carrier_employee`*), in a way that the properties are now understandable by this latest source. In our case, conversion of* `employee` *into* `carrier_employee` *can be enforced through the following mappings:*

- $m_1$: `name` → `fname`;
- $m_2$: `name` → `lname`;
- $m_3$: `organization` → `company`;
- $m_4$: `email` → `email`.

○

The Security Officer is responsible for selecting the mappings to be used for semantic conversion of credential-types, especially when the same property of a credential-type can be mapped into more than one property of another credential-type (e.g., $m_1$ and $m_2$). For the definition of property mappings, one can rely on the semantic content of the names of

| Interactions | Interacting Parties | Initiator | Exchanged Info | Result |
|---|---|---|---|---|
| Subscription | Subject ($s$) and Master source ($S_m$) | $s$ | Info for credentials | • Subscription to $S_m$<br>• X-profile of $s$ |
| Access control | $s$ and $S_m$ | $s$ | Access request ($r$) for target doc.s | • View of the target document(s) in $r$ of all federation sources |
| | $S_m$ and each federated source $S_k$ | $S_m$ | Forwarded access request ($r'$) for target doc.s | • View of the target document(s) in $r'$ of $S_k$ |
| Agreement | $S_m$ and $S_k$ | $S_m$ or $S_k$ | Agreement mode | • New security policies for $S_m$ in each $S_k$ (shallow agreement)<br>• Mappings between credential-types of $S_m$ and those of each $S_k$ (deep agreement) |

**Table 4: Expected interactions in the A2M scheme**

```
<!DOCTYPE employee[
<!ELEMENT employee (name,email,organization)>
<!ELEMENT name (#PCDATA)>
<!ELEMENT email (#PCDATA)>
<!ELEMENT organization (#PCDATA)>
<!ATTLIST employee credID ID #REQUIRED>
<!ATTLIST employee CIssuer CDATA #REQUIRED >
]>
```

**Figure 8: Example of XML credential-type for the A2M deep agreement**

the properties (i.e., tags in the XML specification) using a domain ontology for finding semantically related tags (e.g., synonyms) [6], thus making the semantic conversion process tool-supportable.

## 4.2 Credential-based access control

Credential-based access control to Web documents at a given source can be performed according to two different dissemination modes, namely push and pull. The *push* mode represents the traditional way of performing access control. Under this mode, when a subject needs to access a given document, it submits an explicit access requests to the master source to which it is subscribed. The access request is processed against the $\mathcal{X}$-Sec credential and the policy bases of the master source and, if the A2M scheme is adopted, it is also forwarded to the federation sources. As a result of access control, a `view` is created containing the (portion(s) of) requested documents in the target Web sources the requesting subject is authorized for. If the resulting view is empty the access is denied. Otherwise, the access is granted and the requesting subject is returned the view.

Besides the traditional push mode for performing access control, an additional *push* modality can be succesfully adopted in a web context, suitable for documents that must be released to a large community of subjects and which show a regular behaviour with respect to their release (e.g., they must be periodically distributed or when some pre-defined events happen). According to the push mode, a source periodically broadcasts (portion of) its Web documents to authorized subjects, without the need of an explicit document access request. Also in the push mode, different subjects may be entitled to access different portions of the same document(s). Thus, a way to enforce information push is by encrypting different portions of the same document with different encryption keys [7], and selectively distributing these keys to the various subjects on the basis of the specified ecurity policies. In this way, each subject receives all and only those keys necessary for viewing the portions of the document he/she is authorized to access. Under this mode, an encrypted copy of documents is also maintained by the source.

We refer the interested reader to [2, 3], for a detailed description of the strategies we have developed in the framework of Author-$\mathcal{X}$ to enforce pull and push access control modes.

# 5. RELATED WORK

XML security is a recent research topic and work in this field has concentrated mainly on the development of access control models and encryption techniques. A list of recent papers related to XML security issues can be found at [10]. In particular, papers on access control models here reported borrow some ideas from previous models for object-oriented databases and do not actually take into account some relevant peculiarities of XML. For example, the case of documents not conforming/partially conforming to a DTD is not considered, and no support is provided to the Security Officer for protecting such documents. Moreover, these models only provide the read access mode and do not support credential-based access control.

Original contributions of the Author-$\mathcal{X}$ project are the support for different policies for securing XML documents also in the case of partially and not conforming documents, and the support for a number of specialized access modes for browsing and authoring. Furthermore, Author-$\mathcal{X}$ provides, in addition to the traditional, on user demand mode for document release, a broadcast mode based on a combination of $\mathcal{X}$-Sec security policies and encryption techniques.

Other work related to the topics addressed in this paper is the work on credential specification for stranger parties. In particular, the work by IBM on Trust Policy Language (TPL) [8] is devoted to the enforcement of an XML-based framework for specifying and managing role-based access control in a distributed context where the involved parties are characterized by credentials, and digital certificates are used for party authentication. This framework has been extended for mapping subject certificates to a role, based on policies defined by the owner of the resource and on the roles of the issuers of the certificates [9]. Contributions of those papers more strictly related to our work regard the definition of a uniform XML specification (i.e., the Credential Markup Language) for handling multiple, possibly heterogeneous, certificates. With respect to this, we address credential specification from a complementary point of view, by focusing on an XML-based language specifically conceived for subject credential modeling. As such, our XML credentials could be easily integrated into the framework proposed in [8]. Moreover, our language has also been conceived to specify security policies for Web document access.

# 6. CONCLUDING REMARKS

In this paper, we have presented an XML-based formalism for specifying subject credentials and security policies for protection of Web documents. Furthermore, we discussed the A2O and A2M subscription-based schemes for accessing distributed Web sources based on our proposed credential and policy XML specification. Expressing credentials and security policies using XML has several advantages. First, the protection of Web documents and their security-related information is uniform, in that credentials and policies are XML documents and thus can be protected using the same mechanisms developed for the protection of XML documents. For instance, some credential properties (such as the subject name) may be made accessible to everyone, whereas other properties may be visible only to a restricted class of subjects. Additionally, the use of an XML formalism for specifying credentials facilitates credential submission and distribution. Future research work will be devoted to investigating the following issues.

- Extension of the A2O and A2M schemes for considering a distributed scenario where a subject is not required to subscribe a Web source in order to access its documents. In such a scenario, the access could take place directly by presenting subject credentials according to, for instance, a negotiation approach. Following a decentralized approach, a source may be locally responsible for accepting credentials presented by a subject. Alternatively, in a centralized approach, the source may refer to a third party for getting the appropriate certificate for the requesting subject. To take into account these requirements, a negotiation-based will be developed. Issues related to the development of a negotiation-based scheme are part of the trust management problem in distributed systems, for which some techniques have been proposed in the literature [4, 14, 13]. In this respect, we note that using XML to express security policies makes easier the export of such information when a document migrates from a source to another.
- Integration of the $\mathcal{X}$-Sec credential-based access control with conventional certificate management approaches, such as X.509 [12]. We will investigate how to incorporate our credential specifications into digital certificates, so that we can exploit the authentication services provided by PKIs.

**Acknowledgement**. The authors wish to thank Marco Mesiti for useful preliminary discussions on credentials and XML.

# 7. REFERENCES

[1] E. Bertino, S. Castano, E. Ferrari and M. Mesiti. Specifying and Enforcing Access Control Policies for XML Document Sources. *World Wide Web Journal*, Baltzer Science Publishers, 3(3), 2000.

[2] E. Bertino, S. Castano, and E. Ferrari. Securing XML Documents: the Author-X Project Demonstration. In *Proc. of the SIGMOD 2001 Conferece*, Santa Barbara (CA), May 2001.

[3] E. Bertino, S. Castano, and E. Ferrari. *Author-$\mathcal{X}$*: a Comprehensive System for Securing XML Documents. Techical Report, DSI - University of Milano, submitted for publication.

[4] M. Blaze, J. Feigenbaum, J. Lacy. Decentralized Trust Management. *IEEE Conf. on Security and Privacy*, Oakland, CA, May, 1996.

[5] S. Castano, M.G. Fugini, G. Martella, P. Samarati. *Database Security*. Addison-Wesley, 1995.

[6] S. Castano and V. De Antonellis. A Discovery-based Approach to Database Ontology Design. *Distributed and Parallel Databases – Special Issue on Ontologies and Databases*, 7(1), 1999.

[7] H. Gladney and J. Lotspiech. Safeguarding Digital Library Contents and Users: Assuring Convenient Security and Data Quality. *D-lib Magazine*, May 1997.

[8] A. Herzberg, Y. Mass. Relying Party Credentials Framework. in Proc. of *RSA Conference*, San Francisco, CA, April 2001.

[9] A. Herzberg, Y. Mass, J. Mihaeli. Access Control Meets Public Key Infrastructure, Or: Assigning Roles to Strangers. in Proc. of *IEEE Symposium on Security and Privacy*, Oakland, CA, May, 2000.

[10] C. Geuer Pollmann. The XML Security Page. http://www.nue.et-inf.uni-siegen.de/g̃euer-pollmann/xml_security.html

[11] J. Park, R. Sandhu and G.J. Ahn. Secure Attribute Services on the Web. *ACM TISSEC* (to appear), 2000.

[12] W. Stallings. *Network Security Essentials: Applications and Standars*. Prentice Hall, 2000.

[13] W. Winsborough, K. Seamons, V. Jones. Automated Trust Negotiation. *DARPA Information Survivability Conference and Exposition (DISCEX'2000)*, January, 2000.

[14] M. Winslett, N. Ching, V. Jones, I. Slepchin. Using Digital Credentials on the World Wide Web. *Journal of Computer Security*, 7, 1997.

[15] Word Wide Web Consortium. XML Path Language (Xpath), 1.0, 1999. W3C Recommendation. Available at `http://www.w3.org/TR/xpath`.

[16] Word Wide Web Consortium. Extensible Markup Language (XML) 1.0, 1998. Available at `http://www.w3.org/TR/REC-xml`