Redefining Attributes in XACML to Support Relationship-Based Access Control Policies

Mohammad Jafari, Edmond Scientific Company
mjafari@edmondsci.com

1 Introduction

todo.

2 Abbreviations

To avoid lengthy XML code, I use the following abbreviations and redact XML namespaces where the context is clear.

`AD`	`AttributeDesignator`
`AId`	`AttributeId`
`AV`	`AttributeValue`
`At`	`Attribute`
`DT`	`DataType`

3 Everything is an Attribute

- I assume a simple model in which all access control information are *attributes*.
- An *attribute* assigns a value with a data type to a tuple including a string identifier, and one or more other attributes [1].
- A special λ attribute is defined and its called the null attribute. Attributes defined over λ are simply an identifier assigned to a data type and value.
- A data types can be either one of basic types defined in XACML specifications, an Attribute data type, or a *bag* of other data types.

Semi-formally, this can be written as

$$\begin{aligned} \text{ATT} &::= \lambda \mid \langle \text{id}, \text{DT}, \text{ATT}^n \rangle \mapsto \textit{value} \ (n \geq 1) \\ \text{DT} &::= \text{bag\_of(DT)} \mid \text{attribute} \mid \text{BASIC\_TYPE} \end{aligned}$$

3.1 XML Encoding

To encode this in an XML format, I go by the following rules:

- Cascading attributes and their values can be encoded in a tree structure.

[1] This replaces the `Categories` in XACML 3.0

- `AttributeOf` can be used to link an attribute to other attributes over which it is defined. This accepts a reference to the attributes based on the attribute IDs. XPath-like slash-separated naming can be used to resolve ambiguities.

  See the examples below.

3.2 Examples

3.2.1 User ID

- User ID is an attribute with the identifier "user-id", and data type String, which is defined over the attribute Subject.
- Subject is an attribute with the identifier "subject", and data type Attribute defined over the attribute Request.
- Request is an attribute with the identifier "request" and data type Attribute defined over λ.

The following code shows a sample request context and the Match clause for "if User ID equals John Doe":

```
<At AId="request" DataType="At" >
 <At AId="subject" DT="At" >
  <At AId="subject-id" DT="String">
   <AV>John Doe</AV>
  </At>
 </At>
 <At AId="resource" DT="At" >
  <At AId="resource-id" DT="String">
   <AV>1002120</AV>
  </At>
  <At AId="owner" DT="At">
   <At AId="owner-id" DT="String">
    <AV>John Doe</AV>
   </At>
  </At>
 </At>
 <At AId="action" DT="At">
  <At AId="action-id" DT="String">
   <AV>read</AV>
  </At>
 </At>
</At>
```

```
<Match MatchId="string-equal">
 <AV DT=String>John Doe</AV>
 <AD AId="id" DT="String">
  <AD AId="subject" DT="At" >
   <AD AId="request" DT="At" />
  </AD>
 </AD>
</Match>
```

3.2.2 Owner's Rights

As another simple example, consider the case for granting a resource owner some rights over her/his resource. The following code shows a sample request and a match clause for "if the subject is the resource owner", i.e. subject ID is equal to the resource owner's ID:

```
<At AId="request" DataType="At" >
 <At AId="subject" DT="At" >
  <At AId="subject-id" DT="String">
   <AV>John Doe</AV>
  </At>
 </At>
 <At AId="resource" DT="At" >
  <At AId="resource-id" DT="String">
   <AV>1002120</AV>
  </At>
  <At AId="owner" DT="At">
   <At AId="user-id" DT="String">
    <AV>John Doe</AV>
   </At>
  </At>
 </At>
 <At AId="action" DT="At">
  <At AId="action-id" DT="String">
   <AV>read</AV>
  </At>
 </At>
</At>
```

```
<Match MatchId="string-equal">
 <AD AId="user-id" DT="String">
  <AD AId="owner" DT="At" >
   <AD AId="resource" DT="At" >
    <AD AId="request" DT="At" />
   </AD>
  </AD>
 </AD>
 <AD AId="subject-id" DT="String">
  <AD AId="subject" DT="At" >
   <AD AId="request" DT="At" />
  </AD>
 </AD>
</Match>
```

3.3 Attributes of Multiple Attributes

An attribute can be defined over multiple attributes. Theoretically, an attribute can be defined over any number of other attributes.

3.4 Examples

3.4.1 Next of Kin

It is a typical policy in a healthcare system to allow the patient's next of kin some access to her/his health record. The Boolean attribute *next of kin* is defined over two other attributes: subject and the resource owner. The match clause for "if the subject is the resource owner's next of kin" is shown below:

```
<At AId="request" DataType="At" >
 <At AId="subject" DT="At" >
  <At AId="subject-id" DT="String">
   <AV>John Doe</AV>
  </At>
 </At>
 <At AId="resource" DT="At" >
  <At AId="resource-id" DT="String">
   <AV>1002120</AV>
  </At>
  <At AId="owner" DT="At">
   <At AId="user-id" DT="String">
    <AV>Jane Doe</AV>
   </At>
  </At>
 </At>
 <At AId="action" DT="At">
  <At AId="action-id" DT="String">
   <AV>read</AV>
  </At>
 </At>
</At>
<At AId="next-of-kin" DataType="Boolean" >
 <AOf AId="subject" />
 <AOf AId="resource/owner" />
 <AV>true</AV>
</At>
```

```
<Match MatchId="boolean-equal">
 <AV DT=Boolean>true</AV>
 <AD AId="next-of-kin" DT="Boolean">
  <AD AId="owner" DT="At" >
   <AD AId="resource" DT="At" >
    <AD AId="request" DT="At" />
   </AD>
  </AD>
  <AD AId="subject" DT="At" >
   <AD AId="request" DT="At" />
  </AD>
 </AD>
</Match>
```

3.5 Owner's Rights: Revisited

An alternative way to define owner's rights is to define owner as an attribute over the attributes *resource* and *subject*.

```
<At AId="request" DataType="At" >
 <At AId="subject" DT="At" >
  <At AId="subject-id" DT="String">
   <AV>John Doe</AV>
  </At>
 </At>
 <At AId="resource" DT="At" >
  <At AId="resource-id" DT="String">
   <AV>1002120</AV>
  </At>
 </At>
 <At AId="action" DT="At">
  <At AId="action-id" DT="String">
   <AV>read</AV>
  </At>
 </At>
</At>
<At AId="owner" DataType="Boolean" >
 <AOf AId="subject" />
 <AOf AId="resource" />
 <AV>true</AV>
</At>
```

```
<Match MatchId="boolean-equal">
 <AV DT=Boolean>true</AV>
 <AD AId="owner" DT="Boolean">
  <AD AId="resource" DT="At" >
   <AD AId="request" DT="At" />
  </AD>
  <AD AId="subject" DT="At" >
   <AD AId="request" DT="At" />
  </AD>
 </AD>
</Match>
```

3.5.1 Old Friends

In a social network, a user may want to restrict access to certain resources such as a photo album to her/his closer friends. A match clause for "if the subject is the resource owner's friend at least for 5 years" is shown below. Here, *friend* is a Boolean attribute defined over subject and resource/owner, and it has an integer attribute of its own that represents the length of the friendship.

```
<At AId="request" DataType="At" >
 <At AId="subject" DT="At" >
  <At AId="subject-id" DT="String">
   <AV>John Doe</AV>
  </At>
 </At>
 <At AId="resource" DT="At" >
  <At AId="resource-id" DT="String">
   <AV>1002120</AV>
  </At>
  <At AId="owner" DT="At">
   <At AId="user-id" DT="String">
    <AV>Jane Doe</AV>
   </At>
  </At>
 </At>
 <At AId="action" DT="At">
  <At AId="action-id" DT="String">
   <AV>read</AV>
  </At>
 </At>
</At>
<At AId="friend" DataType="Boolean" >
 <AOf AId="subject" />
 <AOf AId="resource/owner" />
 <At AId="length" DataType="Integer" >
  <AV>6</AV>
 </At>
 <AV>true</AV>
</At>
```

```
<Match MatchId="integer-greater-than-or-equal">
 <AD AId="length" DT="Integer">
  <AD AId="friend" DT="At">
   <AD AId="owner" DT="At" >
    <AD AId="resource" DT="At" >
     <AD AId="request" DT="At" />
    </AD>
   </AD>
   <AD AId="subject" DT="At" >
    <AD AId="request" DT="At" />
   </AD>
  </AD>
 </AD>
 <AV DT=Integer>5</AV>
</Match>
```

4 Attribute over an Attribute Bag

An attribute of type bag is a mechanism to define a list of attributes with the same identifier (distinguished by an index). The index must be a unique string used as a key to retrieve values from the bag.

An attribute defined over a bag is interpreted as an attribute over every single attribute in the bag [2].

4.1 Examples

The following shows the request context and function clause for "if the subject is among the people tagged in the photo". This can also be implemented by defining a *tagged* attribute over the resource and subject, similar to the case discussed above.

```
<At AId="request" DataType="At" >
 <At AId="subject" DT="At" >
  <At AId="subject-id" DT="String">
   <AV>John Doe</AV>
  </At>
 </At>
 <At AId="resource" DT="At" >
  <At AId="resource-id" DT="String">
   <AV>1002120</AV>
  </At>
  <At AId="tagged-users" DT="AtBag">
    <At AId="tagged-users" index="0" DT="At">
     <At Aid="user-id" DT="String">
      <AV>John Doe</AV>
     </At>
    </At>
    <At AId="tagged-users" index="1" DT="At">
     <At Aid="user-id" DT="String">
      <AV>Jane Doe</AV>
     </At>
    </At>
  </At>
 </At>
 <At AId="action" DT="At">
  <At AId="action-id" DT="String">
   <AV>comment</AV>
  </At>
 </At>
</At>
```

```
<Apply FunctionId="any-of">
 <Function FunctionId="string-equal" DT="At" />
 <AD AId="subject-id" DT="At" >
  <AD AId="subject" DT="At" >
   <AD AId="request" DT="At" />
  </AD>
 </AD>
 <AD AId="user-id" DT="String" >
  <AD AId="tagged-users" DT="AtBag" >
   <AD AId="resource" DT="At" >
    <AD AId="request" DT="At" />
   </AD>
  </AD>
 </AD>
</Apply>
```

As a more complex example consider the case of a social network where a user wants to hid a photo album from people who are friends with any of her/his ex-partners. The following shows the request context and the function clause for "if the subject if friend with any of resource owner's ex-spouses".

[2] This can be extended to also support an attribute over the whole bag which will require an extra parameter to specify the type of attribute. Too keep it simple I have avoided this for the moment.

```
<At AId="request" DataType="At" >
 <At AId="subject" DT="At" >
  <At AId="subject-id" DT="String">
   <AV>John F. Kennedy</AV>
  </At>
 </At>
 <At AId="resource" DT="At" >
  <At AId="resource-id" DT="String">
   <AV>1002120</AV>
  </At>
  <At AId="owner" DT="At">
   <At AId="user-id" DT="String">
    <AV>Arthur Miller</AV>
   </At>
   <At AId="ex-partners" DT="AtBag">
    <At AId="ex-partners" index="0" DT="At">
     <At AId="user-id" DT="String">
      <AV>Mary Slattery<AV>
     </At>
    </At>
    <At AId="ex-partners" index="1" DT="At">
     <At AId="user-id" DT="String">
      <AV>Norma Jeane Mortenson<AV>
     </At>
    </At>
    <At AId="ex-partners" index="2" DT="At">
     <At AId="user-id" DT="String">
      <AV>Inge Morath<AV>
     </At>
    </At>
   </At>
  </At>
 </At>
 <At AId="action" DT="At">
  <At AId="action-id" DT="String">
   <AV>read</AV>
  </At>
 </At>
</At>
<At AId="friend" DataType="Boolean" >
 <AOf AId="request/subject" />
 <AOf AId="request/resource/owner/ex-partners" index="1" />
 <AV>true</AV>
</At>
```

```
<Apply FunctionId="boolean-is-in">
 <AV DT="Boolean">true</AV>
 <AD AId="friend" DT="Boolean" >
  <AD AId="subject" DT="At" >
   <AD AId="request" DT="At" />
  </AD>
  <AD AId="ex-partners" DT="AtBag" >
   <AD AId="owner" DT="At" >
    <AD AId="resource" DT="At" >
     <AD AId="request" DT="At" />
    </AD>
   </AD>
  </AD>
 </AD>
</Apply>
```

5 Additional Features

5.1 Symmetry

todo.

5.2 Inverse

todo.

5.3 Diameter and Transitive Closure

todo.