A Privacy Oriented Attribute Exchange in Shibboleth

Grid-Middleware Workshop, 23rd APAN Meeting, Manila, Philippines, Jan. 25, 2007

Shoichirou FUJIWARA†
Takaaki KOMURA†† Yasuo OKABE††

† : Graduate School of Informatics, Kyoto Univ.
†† : Academic Center for Computing and Media Studies, Kyoto Univ.
Agenda

- Outline of our work: extending attribute exchange to enhance privacy in Shibboleth
- Design and implementation
- Summary and future works
Background

- Necessary processes and a deployment framework for Web services

- **Identity**: Information required for AuthN/AuthZ
- **Shibboleth**: a project of Internet2/MACE
What we did: Extension of Attribute Exchange in Shibboleth

- Current protocol
  - Tell me the user’s “Grade”
  - One’s “Grade” = “M2”
  - Authorize users in case their “Grade” $\geq$ “B4”

- Privacy risk: attribute information for the SP is often unnecessarily detailed
What we did: Extension of Attribute Exchange in Shibboleth

- Current protocol
  
  ![Diagram](Diagram.png)

- Privacy risk: attribute information for the SP is often unnecessarily detailed

- Proposal: SP presents AuthZ conditions
  
  ![Diagram](Diagram.png)
Agenda

- Outline of our work: extending attribute exchange to enhance privacy in Shibboleth
- Design and implementation
- Summary and future works
Protocol Message Extension (Request from SPs to IdPs)

- Existing attribute request
  
- Extended request

Request → Attribute Query → Attribute Designator(s)

Request → Condition Evaluation Query → Condition(s) [described later]
Protocol Message Extension (Response from IdPs to SPs)

- Existing response to attribute request
  - Response → Assertion(s) → Attribute Statement(s)
- Extended response
  - Response → Assertion(s) → Attribute Statement(s) → Condition Evaluation Statement(s) (described later)
Protocol in detail

User

IdP

Configurations (application level)

User Front-end (Apache module)

Attribute Requester (cache assertions)

by RPC

AuthN Component

AuthZ Component

AuthN Component

AuthZ Component
Protocol in detail

**User**

1. **User Front-end (Apache module)**

**IdP**

**AuthN Component**

**AuthZ Component**

**SP**

**Configurations (application level)**

**Attribute Requester (cache assertions)**

by RPC
Protocol in detail

Configurations (application level)

User Front-end (Apache module)

Attribute Requester (cache assertions)

AuthN Component

AuthZ Component

1. User 

2. IdP
Protocol in detail

- **IdP**
- **SP**
- **User**
- **Attribute Requester** (cache assertions)
- **User Front-end** (Apache module)
- **Configurations** (application level)
- **AuthN Component**
- **AuthZ Component**

1. User communicates with the User Front-end (Apache module).
2. The User Front-end requests configurations from the IdP.
3. The IdP responds with AuthN and AuthZ Component configurations.
Protocol in detail

Configurations (application level)

User Front-end (Apache module)

Attribute Requester (cache assertions)

AuthN Component

AuthZ Component
Protocol in detail

User

1. User
2. AuthN Component
3. AuthZ Component
4. User Front-end (Apache module)
5. Attribute Requester (cache assertions)

Configurations (application level)

SP

IdP
Protocol in detail

Send a **Condition Evaluation Query**, the conditions for authorizing the user, instead of an **Attribute Query**

NOTE: The sent conditions varies per Application, not URI
Extract conditions and Attribute Designators from the Query, then evaluate the conditions and get the releasable attributes written in the designators, and return them as Condition Evaluation Statements and Attribute Statements.
Protocol in detail

The statements sent from the IdP

Configurations (application level)

User Front-end (Apache module)

Attribute Requester (cache assertions)

AuthN Component

AuthZ Component
Authorize the user to the resource based on the statements

NOTE: the access control rules vary per URI, not Application
Language for Describing the Conditions

- **Points**
  - What to describe: AuthZ conditions about attributes
  - Compactness: XACML is too complicated

- **Design**
  - Syntax:
    - **Data types:** string and long integer only
Language for Describing the Conditions

- **Points**
  - What to describe: AuthZ conditions about attributes
  - Compactness: XACML is too complicated

- **Design**
  - Syntax:
    - Data types: string and long integer only

- **Predicate**
  - “=”, “≥”, etc.

- **Logic Function**
  - “AND”/”OR”/”NOT”

- **Immediate Value**

- **Attribute Designator**

- **Condition**
  - Cond. ID

- **Needless?**
Evaluation and its Result

- **Evaluation**: at IdPs

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation:** at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation: at IdPs**
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

```
Condition >> AND >> Grade

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
```
Evaluation and its Result

- **Evaluation: at IdPs**
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

**Condition**: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

\[
\text{Condition: \text{"Grade"} > \text{"B4"} \& \text{first 3 digits of "phone no." = "753"}}
\]

\[
\text{Condition: true AND (\text{Grade} > \text{"B4"} AND \text{phone no.} = \text{"753"})}
\]
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
**Evaluation and its Result**

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

**Evaluation**: at IdPs
- “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**: at IdPs
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

Condition: “Grade” > “B4” & first 3 digits of “phone no.” = “753”
Evaluation and its Result

- **Evaluation**
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

- **Result**

  - unanswerable

  - user-policy
Evaluation and its Result

- **Evaluation**
  - “Grade” is “M2” and “phone no.” is not released due to user’s policy.

- **Result**
  - Sent to the SP

Condition

- **Evaluation Statement**

- **Evaluation Result**
  - ID="1234"
  - user-policy
  - unanswerable
“Reasons” of “Unanswerable”

- To be ‘polite’ to users
- An IdP attaches just one “reason” to a evaluation result of “unanswerable”
- 4 Kinds of “reasons” of “unanswerable”

<table>
<thead>
<tr>
<th>Strong</th>
<th>Processing Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Admin Policy</td>
</tr>
<tr>
<td></td>
<td>Attributes Not Found</td>
</tr>
<tr>
<td>Weak</td>
<td>User Policy</td>
</tr>
</tbody>
</table>
Calculation of “Reasons” @IdP

- Immediate Value: has no reasons
- Attribute Designator: calculates reasons based on ARPs actually applied for filtering if no values are obtained.

- Operation: (in case of “undetermined”)

  ![Diagram]

  - An Operation
    - undetd
      - Reason: A, B
    - undetd
      - Reason: A, C
    - undetd
      - Reason: A, B, C

- Predicate: (in case of “unanswerable”)
  - As well as the case of “Operation”
Calculation of “Reasons” @IdP

- Logic Function (in case of “unanswerable”)
  - AND: “the strongest reason” in its operands
  - OR: “the weakest reason” in its operands
  - NOT: reasons of its operand

- Condition (in case of “unanswerable”): “the strongest reason” among its child node
Coping with Multivalued Attributes (to be implemented)

- Specify preference of each “Predicate” as to “true/false”
- Adopt the attribute value with which the predicate is evaluated as the most desirable value.
Access Control @SP

- Example

- Processing: in a similar way by which the condition language processed at IdPs

“unanswerable” with its reason → “deny” with the reason
Handling “unanswerable” (“deny”) According to “reason”

The reason of denial is as follows:

Some attributes required for the authorization decision aren’t released to us, and this is because of your attribute release policy. So, you need to change it.

Based on the information about you, you are not resource at “https://s

Please contact the administrator of this service or application if you believe this to be an error.
“Reasons” of “Deny” @SP

12 Kinds

- ADMIN_POLICY
- PRED_RULE
- NORMAL_RULE
- UNKNOWN
- SOME_ERROR_SP
- SOME_ERROR_IDP
- INVALID_RULE
- PRED_NOT_SENT
- ATTR_NOT_FOUND
- USER_POLICY
- NO_SESSION
- GRANTED

IdP > SP > user (-side reasons)
Definite “deny” > error > any shortages
Implementation

- Platform: Apache 2.0.58/Tomcat 5.5.20/Java 2 SDK 1.5.0 on Fedora Core 4
- IdP: Opensaml 1.1b, Shibboleth 1.3c
- SP: Opensaml 1.1a, Shibboleth 1.3f (C++, source code build)
Demonstration

- A user m1 tries to access a DEMO book rental service at an SP
  - AuthZ condition: user’s grade $\geq$ B4
- 1<sup>st</sup> trial: denied due to one of the user’s policies
  - change the policy
- 2<sup>nd</sup> trial: successfully authorized
Agenda

- Outline of our work: extending attribute exchange to enhance privacy in Shibboleth
- Design and implementation
- **Summary and future works**
Summary

● An Extended protocol of attribute exchange in shibboleth to enhance user’s privacy
  ● An SP to an IdP: conditions for authorization
  ● The IdP to the SP: their evaluation result with reasons why the IdP evaluated so

● How to control access requests
  ● Utilizing reasons of “unanswerable” and “deny”

● Implementation and Demo.
Future Works

- Policy filtering
- Application to Shibboleth 2.0
- Verification of practicality
- Blend of the existing strategy of attribute release at IdPs and that of our extension
Thank you for your attention! Questions?