Distributed Access Control Management in Federated Identity Systems

Manuel Gil Pérez[1], Gabriel López[1], Antonio F. Gómez Skarmeta[1],
Alberto Sicre Vara-De-Rey[2] and Aljosa Pasic[2]

[1] Departamento de Ingeniería de la Información y las Comunicaciones
University of Murcia, Spain
Email: {mgilperez,gabilm,skarmeta}@um.es
[2] ATOS Origin, Albarracin 25, 28037 Madrid
Email: {alberto.sicre,aljosa.pasic}@atosresearch.eu

Abstract. Identity federation provides a powerful way for managing sensible information of its users. However, as the number of members increases, the management of the policies defined by the federation is becoming more and more complex. In this paper, we present a mechanism to manage this complexity by means of the *administrative delegation*. This allows administrators of an institution to delegate part of these policies to other users, who also will have more knowledge within the scope where these policies will be enforced. This proposal also introduces a way for shortening the added complexity that supposes the introduction of this new sort of policies to users without deep knowledge in the policy management area.

Key words: administrative delegation, delegation policies, access control, distributed authorization

1 Introduction and Motivation

As the number of members of an institution increases, new institutions join the same federation, or the policies defined in a federation change due to the highly dynamic nature of this sort of systems, the management of their policies becomes more and more complex. This is mainly due to the great amount of policies that should be managed, such as access control policies, privacy policies, or validation policies based on LoA (*Level of Assurance*), among others.

In order to reduce this complexity, the system administrator of an institution can delegate to third parties, called *delegates*, the management of a subset of the system policies. Thus, we are not only distributing the management of those policies to other people, but also they are being delegated to those who have more knowledge in the application area where they will be used. This process, where the administrator transfers the management of a subset of policies to a delegated person, is commonly known as *administrative delegation* [1, 2].

The introduction of this new sort of policies supposes a new value-added service for the current identity systems, but also presents some drawbacks that have to be treated suitably:

- The number of policies to manage increases, so that the system administrator will have to manage both the policies that already existed previously (access control policies to services and resources, privacy policies, etc.) and this new sort of policies.
- The delegates are usually users with no knowledge in the policy management, access control languages such as XACML, etc. Therefore, we will have to make easier to these people the generation and management of this new sort of policies, and make it as easy and intuitive as possible.

As we can see, even though the workload of the administrator is reduced, and considerably distributed among several delegates, the policy management (including the administrative ones) will also be more complex. Therefore, it is not enough to define policies for the administrative delegation, but also it will be necessary to define an infrastructure that can manage these policies, thereby helping, mainly to the delegates, to carry out these tasks.

2 Application Scenario

As application example of the administrative delegation in real environments, let us suppose a scenario where the administrator of an institution delegates to all heads of department the policies about granting access to those people under their supervision. These heads of department will have more knowledge about their own employees than the system administrator. In this scenario, the system administrator will be able to delegate in each head of department the definition of which employees will have access to the network, as well as the connection schedule according to their workday; that information is perfectly known by each head of department.

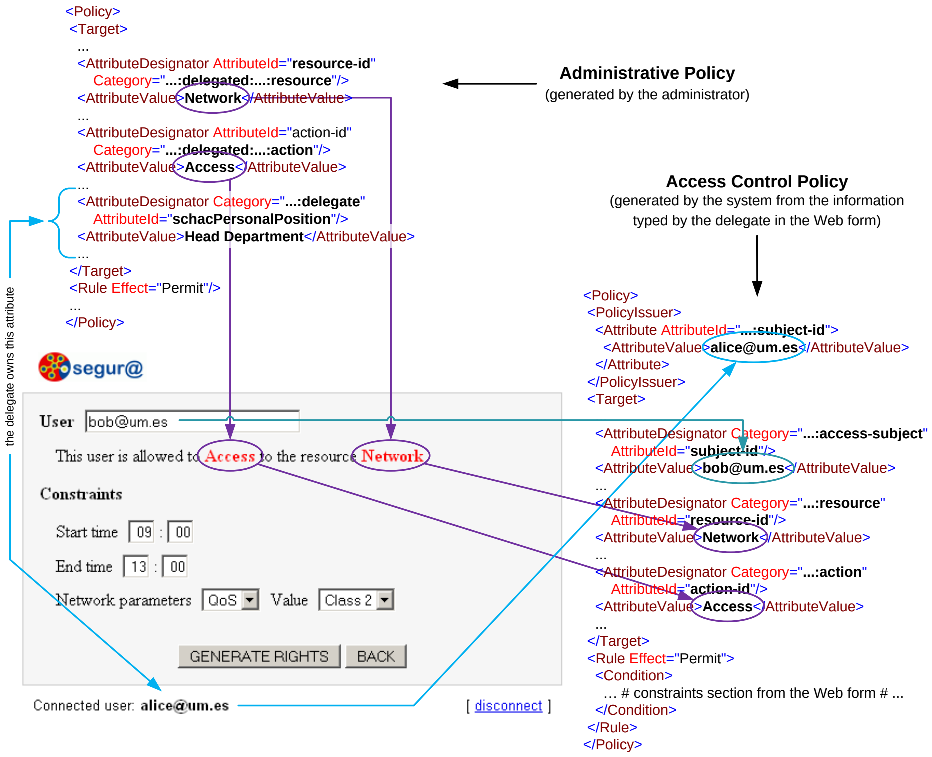
In this example scenario we can see the use of the administrative delegation, where we are avoiding that the system administrator has to create access control policies on a set of people that he (probably) does not know. In this case, each head of department, once the system administrator authorizes him as a delegate, will be the person in charge of controlling the access to the network of his employees by creating the needed access policies with the adequate information.

3 Delegation Policy Management

As a solution to the problems previously commented, we have included in this work a set of new components to the Segur@-DAMe identity federation [3] for managing the complete life cycle of these new policies. We have also defined a mechanism by which we allow generating a set of templates (Web forms) to help delegates to perform these administrative tasks in a simple and intuitive way. These templates are automatically generated by the infrastructure from the administrative delegation policies created by the system administrator.

To this end, our infrastructure makes use of XACML 3.0 (*eXtensible Access Control Markup Language*) [4], which includes in its specification new advanced features for the definition of delegation policies. These changes have been made to allow any person of an institution, who owns a certain privilege, to delegate it to another person.

In this sense, XACML 3.0 defines a new element, called *PolicyIssuer*, to indicate who has issued a given policy. With this element, the system can identify and verify whether the corresponding issuer is valid to delegate the enclosed privilege just before the policy is used. A policy with no issuer element is considered as trusted and, therefore, will be managed by the PDP (*Policy Decision Point*) as a traditional policy.



Administrative Policy
(generated by the administrator)
Access Control Policy
(generated by the system from the information typed by the delegate in the Web form)
the delegate owns this attribute
segur@

```
<Policy>
 <Target>
  ...
  <AttributeDesignator AttributeId="resource-id"
     Category="...:delegated:...:resource"/>
  <AttributeValue>Network</AttributeValue>
  ...
  <AttributeDesignator AttributeId="action-id"
     Category="...:delegated:...:action"/>
  <AttributeValue>Access</AttributeValue>
  ...
  <AttributeDesignator Category="...:delegate"
     AttributeId="schacPersonalPosition"/>
  <AttributeValue>Head Department</AttributeValue>
  ...
 </Target>
 <Rule Effect="Permit"/>
 ...
</Policy>
```

User bob@um.es

This user is allowed to Access to the resource Network

Constraints

Start time 09 : 00

End time 13 : 00

Network parameters QoS ▾ Value Class 2 ▾

Connected user: **alice@um.es** [disconnect]

```
<Policy>
 <PolicyIssuer>
  <Attribute AttributeId="...:subject-id">
   <AttributeValue>alice@um.es</AttributeValue>
  </Attribute>
 </PolicyIssuer>
 <Target>
  ...
  <AttributeDesignator Category="...:access-subject"
     AttributeId="subject-id"/>
  <AttributeValue>bob@um.es</AttributeValue>
  ...
  <AttributeDesignator Category="...:resource"
     AttributeId="resource-id"/>
  <AttributeValue>Network</AttributeValue>
  ...
  <AttributeDesignator Category="...:action"
     AttributeId="action-id"/>
  <AttributeValue>Access</AttributeValue>
  ...
 </Target>
 <Rule Effect="Permit">
  <Condition>
   … # constraints section from the Web form # ...
  </Condition>
 </Rule>
</Policy>
```

Fig. 1. Automatic generation of templates and access control policies

As we have commented before, the main goal of our infrastructure is to provide delegates with an easy and intuitive way for creating the access control policies for which they are responsible. For this purpose, we make use of a PMT (*Policy Management Tool*) which is capable of parsing administrative policies and extracting the information necessary to generate the corresponding templates in an automatic fashion. The PMT will then make use of XSL transformations [5] to carry out this task, taking as input the administrative policy generated by the administrator. The rest of information that cannot be directly found in that policy is requested to the delegate as input fields in the template.

Fig. 1 depicts this process, in a schematic way, for: 1) generating the templates (Web forms) from the administrative policy created by the system administrator (top of the image); and 2) generating the final access control policies from some pieces of the administrative policy and the information provided by the delegate in the Web form (right-hand size of the image). Both processes are automatically carried out by making use of the XSL transformations mentioned above.

4 Conclusion

As we have seen throughout this paper, we have presented a new way for managing the administrative delegation in which system administrators or any institution, belong to the same identity federation, can delegate part of their work to third parties. With the help of this system, the policy management is being distributed to those people who have more knowledge than the system administrator on the scope where those policies will be applied later.

To this end, our infrastructure is capable of automatically generating a set of templates (or Web forms) from the administrative policies created by the administrators. These delegates, who have (probably) no idea about how to create policies, will be able to fill in these templates in an easy and intuitive way.

Acknowledgment

This work has been funded by the CENIT Segur@ (Seguridad y Confianza en la Sociedad de la Información) project. Authors would also like to thank the Funding Program for Research Groups of Excellence with code 04552/GERM/06 granted by the Fundación Séneca.

References

1. E. Rissanen and B.S. Firozabadi. “Administrative Delegation in XACML - Position Paper”. Swedish Institute of Computer Science, September 2004.
2. K. Gaaloul and F. Charoy. “Task Delegation Based Access Control Models for Workflow Systems”. In *I3E ’09: Proceedings of the 9th IFIP Conference on e-Business, e-Services and e-Society*, pages 400–414, 2009.
3. The CENIT Segur@ Project. “Seguridad y Confianza en la Sociedad de la Información”. http://www.cenitsegura.com.
4. E. Rissanen (editor). “XACML v3.0 Administration and Delegation Profile Version 1.0”. Committee Draft 01, April 2009.
5. M. Kay (editor). “XSL Transformations (XSLT) Version 2.0”. W3C Recommendation, January 2007.