A survey on authorization and delegation

Isaac Agudo

Abstract: *The paper presents a survey of actual solutions for authorization and delegation. It reviews both academicals and enterprise solutions, remarking their strong and weak points*

Key words: *Computer Systems and Technologies, Authorization, Delegation, Federation, Single Sing On*

INTRODUCTION

Although authorization and delegation are two concepts of main interest in any organization, only a few solutions have been implemented to provide both authorization and delegation in a distributed way. In such organizations, were resources are spread, like distributed data bases or web services, there is a need for architectures to support authorization and delegation. This allows individuals to deny or grant access to their resources, defining authorization policies. Authorization is usually divided in two phases, firstly authentication and secondly access control, but there is a tendency to merge these two phases and define authorization as an atomic concept.

AUTHORIZATION SCHEMES

In this section we will analyze some of the most interesting authorization schemes that have been proposed in the literature so far. In fact, and because of the many solutions that have can be found on this topic, we will mainly focus on those ones that have been supported by international bodies and organizations, or that have special implications in commercial products in the information security market. In the different subsections, we will review each of the solutions.

PolicyMaker and Keynote

PolicyMaker is a general and powerful solution that allows the use of any programming language to encode the nature of the authority being granted as well as the entities to which it is being granted. Keynote is a derivation of PolicyMaker, and has been supported by IETF.

Blaze, Feigenbaum and Lacy introduced in [4] the notion of Trust Management. In that original work they proposed the PolicyMaker scheme as a solution for trust management purposes. It addresses the authorization problem directly, without considering two different phases (one for authentication and another for access control).

KeyNote (RFC 2704) was proposed and designed to improve two main aspects of PolicyMaker. On the one hand, to achieve standardization and on the other hand, to facilitate integration into applications.

Keynote uses a specific assertion language that is flexible enough to handle the security policies of different applications. Assertions delegate the authorization to perform operations to other principals. KeyNote considers two types of assertions called policies and credentials.

- *Policies*. This type of assertions does not need to be signed because they are locally trusted. They do not contain the corresponding Issuer of PolicyMaker.
- *Credentials.* This type of assertions delegate authorization from the issuer of the credential, or *Authorizer*, to some subjects or *Licensees* (see later for details) if some *Conditions* hold. They have to be signed by the authorizer.

Figure 1 shows an example of assertion. It states that an RSA key 12345678

authorizes the DSA keys abcd1234 1234abcd for read and write access on the database.

```
KeyNote-Version: 2
Authorizer: "rsa-hex:12345678"
Licensees: "dsa-hex:abcd1234" || "dsa-hex:1234abcd"
Comment: Authorizer delegates read and write access
         to either of the licensees
Conditions:  (resource == "database" &&
             (access == "read") || (access == "write"))
Signature: "sig-rsa-md5-hex:abcd1234"
```

Figure 1. KeyNote assertion

SDSI/SPKI

This solution is a unification of two similar proposals, SDSI (Simple Distributed Security Infrastructure) and SPKI (Simple Public Key Infrastructure). SPKI was proposed by the IETF working group and, in particular, by Carl Ellison [2]. SDSI was an alternative design for a public-key infrastructure to X.509, designed by Ronald L. Rivest and Butler Lampson [3].

The SPKI/SDSI certificate format is the result of the SPKI Working Group of the IETF (SPKI Certificate Theory, RFC 2693). The main feature of SDSI/SPKI is that its design provides a simple public key infrastructure which uses linked local name spaces rather than a global, hierarchical one. All entities are considered analogous; hence, every principal can produce signed statements.

The data format chosen for SPKI/SDSI is S-expression. This is a LISP-like parenthesized expression with the limitations that empty lists are not allowed and the first element in any S-expression must be a string, called the "type" of the expression. SDSI establishes four types of certificates: Name/Value, Membership, Autocert and Delegation.

SPKI/SDSI unifies all types of SDSI certificates into one single type of structure. The SPKI/SDSI certificate contains at least an Issuer and a Subject, and it can contain validity conditions, authorization and delegation information. Therefore, there are three categories: ID (mapping <name,key>), Attribute (mapping <authorization,name>), and Authorization (mapping <authorization,key>). Figure 2 details the relationship between key, name and authorization sentences and the three possible SDSI/SPKI certificate categories.

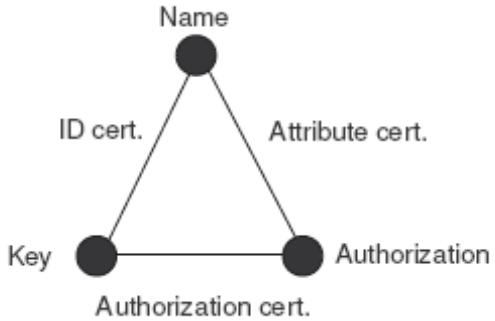

Figure 2: SPKI Certificate Types

Privilege Management Infrastructure (PMI)

The last X.509 ITU-T Recommendation [6] introduces the concept of Privilege Management Infrastructure (PMI) as the framework for the extended use of attribute certificates. The Recommendation establishes four PMI models: (i) General, (ii) Control,

(iii) Roles and (iv) Delegation. The first one can be considered as an abstract model, while the other ones can be considered as the models for implementation. The PMI area inherits many concepts from the Public Key Infrastructure (PKI) area. In this sense, an Attribute Authority (AA) is the authority that assigns privileges to users, and the Source of Authorization (SOA) is the root authority in the delegation chain. A typical PMI will contain a SOA, a number of AAs and a multiplicity of end entities (EE). (RFC3281)

Initially, the Source of Authority assigns or delegates the privilege to Attribute Authorities. These can delegate the privileges to other AAs or to EEs. AAs and EEs can use their delegated privileges and present them to the Privilege Verifier (PV), that verifies the certification path to determine the validity of the privileges. The difference between AA and EE is that EE can not further delegate the privileges to other entities, becoming the leaves of the tree. The PV must trust the SOA in order to verify the certification path, as they may reside in different domains. The mechanism (data structure) used to contain the delegation statement(s) is the attribute certificate. The Extensions field is used by the authorities to include the delegation policy.

FEDERATION SOLUTIONS

In this section we analyze some of the most interesting federation solutions that have been developed by different consortiums or enterprises. We focus on two significant solutions such as Shibboleth, and .Net Passport. These selected solutions represent both educational and enterprise points of view. Shibboleth is the representative for academia solutions, although there are other solution like PAPI and Athens. In the other hand, we chose .Net Passport as the enterprise representative, although their opponent, Liberty Alliance (http://www.liberty.org), is growing in popularity, mainly due to the relevancy of the partners that conforms the consortium.

The general definition of Federation is the act of establishing a trust relationship between two entities, or more detailed an association comprising any number of service providers and identity providers. Therefore, Federation should be understood as delegation of services where the service providers delegate the security management to identity providers

Microsoft Passport

At the end of 90's, as part of its .NET initiative, Microsoft introduced a set of Web services that implement a so-called "user-centric" application model, and that are collectively referred to as .NET My Services. At the core of Microsoft .NET My Services is a password-based user authentication and Single Sign-In service called Microsoft .NET Passport (http://www.passport.com). The fundamental component of a Federation Solution is Single Sign-In (SSI) Service, therefore Microsoft .NET Passport could be considered as the first partial Federation Solution. Microsoft .NET Passport users are uniquely identified with an email address (usually hotmail and MSN accounts) and all participating sites are uniquely identified with their DNS name. Passport use a series of cookies to store the authentication information and to assist the sing-in functionality in the user computer.

Shibboleth

Shibboleth is a project of Internet2/MACE (http://shibboleth.internet2.edu). The purpose of the proposal is typically to determine if a person using a web browser has the permissions to access a target resource based on information such as being a member of an institution or a particular class. It is implemented by using federated administration.

In federated administration usually, a resource provider leaves the administration of user identities and attributes to the user's origin site. Therefore users are registered only at their origin site, but not at each resource provider. Moreover, the system is privacy preserving in the sense that it do not use identity information. Therefore, it is necessary to

associate a handle with the user. This handle stores the security information without exposing the identity of the user.

Consequently, Shibboleth is a system for securely transferring attributes about a user, from the user's origin to a resource provider site. Two principal components are in charge in performing the attribute transference, Attribute Authority (AA) in the user side and Shibboleth Attribute Requester (SHAR) in resource side. These components interchange authorization information by exchanging SAML [1] messages using any shared protocol that supports the required functional characteristics.

SPECIFIC DELEGATION SCHEMES

In this section we will focus on different formalism that have been specifically developed to support delegation services and that can be integrated into a multiplicity of applications. Those schemes will be explained and analyzed, but moreover, we will present regarding how to include the solutions on existing working frameworks, what would facilitate the introduction of users' delegation operations into final applications

Logic frameworks.

Logic programming offers a powerful mechanism to represent authorization and access control decisions [5, 7]. In this context, authorizations are represented as predicates and decisions are based on formulae verification.

There are many solutions for formulae verification but the most known is probably PROLOG [8], which has several implementations for different platforms (Windows, Linux, Macintosh,….). Having this amount of different implementations, most of them provided with some kind of free license, make it easy to implement authorization decision systems based on formulae verification.

Ruan et al proposed in [9] a logic approach to model delegation. They base their approach in extended logic programs, so they allow explicit negation (denial) of authorization. Their language is based on the following concepts:

- *Subjects*. The grantors and grantees of authorizations.
- *Objects*. The target of authorizations.
- *Access rights*. The different ways an object can be accessed.
- *Authorization Type*. DAP considers three authorization types: Negative authorization (-), Positive authorization (+) and Delegatable authorization (*).

A negative authorization specifies the access that must be forbidden, while a positive authorization specifies the access that must be granted. A delegatable authorization specifies the access that must be delegated as well as granted. DAP defines three partial orders $<_S$, $<_O$, $<_A$ to represent inheritance hierarchies of subjects, objects and access rights, respectively. DAP rules are Horn clauses, which are logic predicates of the form $b_0 \leftarrow b_1,\ldots,b_k,not\,b_{k+1},\ldots,not\,b_{k+m}, m \geq 0$ where each b_i is a literal and not is the negation as failure symbol.

Given a DAP, we may ask if a particular authorization predicate p is true, then we should try to infer p from the rules of the DAP. As there are both positive and negative authorizations in a DAP, there could be conflicts among authorization, i.e. contradictory authorization predicates. DAP proposes several methods for solving conflicts.

Contrary to DAP, Role Based Trust Management (RT) proposed by Li [12] does not support negative statements, so RT does not have to worry about conflict resolution. It is based on a subset of Prolog, Datalog [10,11] which is a language of facts and rules. Datalog is a logic based query language for the relational model that has been mainly used in the field of knowledge discovery but also in some other fields. One of the more attractive properties of DATALOG, regarding its tractability, is the absence of function-symbols as arguments in the predicates. It is the main reason for DATALOG to have

efficient procedures for answering queries.

RT makes use of Roles to define a full framework composed by different languages, each of them with different characteristics. Roles can be interpreted as privileges or attributes, and are the analogous for the combination of access right and resource in DAP.

Graph frameworks.

Due to this obscure transcription of previous solutions, there is a need for graphical solutions that cover the gap from the administrator point of view of the system to the logic formalism. Graph based solutions are thought to be less powerful but more expressive and more understandable, as they can be represented graphically. We will explain the pros and cons of graph based reasoning solutions in opposition with logic based reasoning solution

A graphical solution may be based on the use of directed graphs to model authorization and delegation process. Basically, this maps each predicate to a directed arc in a graph. Arcs go from the issuer of the authorization or delegation statement to the subject who is authorized or granted privileges. There are as many different arcs as different authorization/delegation statements to consider. In this way, all the authorization and delegation relationships are represented in the same chart making easier for an inexperienced user to understand how the system is defined. Diagrams are always the first step in the process of software engineering (see UML) and so they are, or they should be, in the field of security and authorization.

Varadharajan and Ruan have proposed two solutions to represent authorization and delegation using directed graphs. In [13] they present a first approach to the problem. This approach considers three types of authorizations: negative authorization, positive authorization and delegatable authorization, a cross line represents a negative authorization, a dashed line represents a positive authorization and a simple line represents a delegatable one. In [14] the same authors proposed a new approach, *weighted graphs*. In that proposal, each authorization is associated with a weight given by the grantor, representing the degrees of certainties about the authorization grants. The weight is a non-negative number, and a smaller number represents a higher certainty. When considering both negative and positive authorizations, we get conflicts if the same subject is issued a negative and a positive authorization. In this case, we need to define a conflict resolution method that allows us to decide which of them has to be considered.

An evolution of these solutions is *Weighted Trust Graph* (WTG), presented in [15], which aims to generalize the previous approaches. In fact, WTG support the previous proposal as a particular case of our framework. Additionally, WTG allows defining more complex policies. Even if in other solutions a delegation statement is usually issued together with an authorization statement, our solution can use both of them separately, allowing us to introduce the notion of negative delegation. We define negative and positive delegation statements as trust on negative and positive authorization, respectively. WTG assign to each authorization a weight that, together with the security level policy, allows avoiding many conflicts. In case the weights are the same, WTG follows a predecessor-take-precedence principle with some refinements; that is, a new conflict resolution method called *strict-predecessor-take-precedence*.

CONCLUSIONS

Although there are several approaches to implement authorization in actual applications, none of them cope with all the expectative of final users. In particular, they manage delegation in different ways. In order to define an standard framework for authorization and delegation we have to merge all the actual initiatives into one unified framework. This framework should implement both a graphical interface and a powerful reasoning mechanism.

REFERENCES

[1] Cantor, S., Kemp, J., Philpott, R., & Maler, E Security Assertion Markup Language (SAML V2.0). OASIS. Retrieved March 15, 2005, from http://docs.oasis-open.org/security/saml/v2.0/

[2] Ellison, C., Frantz, B., & Lacy, J. Simple public key certificate. Internet Draft draft-ietf-spki-cert-structure-06.txt

[3] Rivest, R., & Lampson, B. SDSI -A Simple Distributed Security Infrastructure. Working document, Presented at CRYPTO '96.

[4] Blaze, M., Feigenbaum, J. & Lacy, J. Decentralized Trust Management. In IEEE Symposium on Security and Privacy. IEEE Computer Society Press pp. 164-173.

[5] Barker, S. (2000) Data protection by logic programming. Lecture Notes in Computer Science, 1861:1300-1314.

[6] ITU. (2000) X509.Information technology Open systems interconnection. The Directory: Public-key and attribute certificate frameworks.

[7] Crampton, J., Loizou, G. & O'Shea, G. (2001) A logic of access control. The Computer Journal, 44:54-66.

[8] Nilsson, U. & Maluszynski, J. (2000) Logic, Programming and Prolog (2ed)

[9] Ruan, C., Varadharajan, V. & Zhang, Y. (2002) Logic-based reasoning on delegatable authorizations. In Proc. of the 13th International Symposium on Methodologies for Intelligent Systems, pp 185-193, Springer LNCS 2384.

[10] Ullman, J.D. (1988) Principles of Database and Knowledge-Base Systems, Volume I and II. Computer Science Press 1988

[11] Abiteboul, S. & Hull, R. (1988) "Data functions, datalog and negation," in Proc. ACM-SIGMOD Conf.

[12] Li, N., Mitchell, J.C. & Winsborough, W.H. (2002) Design of a role-based trust management framework. In Proceedings of the 2002 IEEE Symposium on Security and Privacy, pages 114\u2013130. IEEE Computer Society Press.

[13] Ruan, C. & Varadharajan, V. (2003) A formal graph based framework for supporting authorization delegations and conflict resolutions, International Journal of Information Security, Volume 1, Issue 4, Pages 211 - 222

[14] Ruan, C. & Varadharajan, V. (2004), A Weighted Graph Approach to Authorization Delegation and Conflict Resolution, Lecture Notes in Computer Science, Volume 3108, Pages 402 – 413

[15] Agudo, I., Lopez, J. & Montenegro, J.A. (2005) A Representation Model of Trust Relationships with Delegation Extensions, Lecture Notes in Computer Science, Volume 3477, Dec 2005, Pages 116 - 130

ABOUT THE AUTHOR

PhD. Student Isaac Agudo, Department of Computer Systems, University of Malaga, E-mail: isaac@lcc.uma.es.