Available online at www.sciencedirect.com

ScienceDirect

journal homepage: www.elsevier.com/locate/cose

Computers & Security

Survey of certificate usage in distributed access control

Yki Kortesniemi*, Mikko Särelä

Aalto University, Finland

ARTICLE INFO

Article history:
Received 23 July 2013
Received in revised form
6 March 2014
Accepted 30 March 2014

Keywords:
Access control
Authorisation
Certificate
SPKI
Distributed systems

ABSTRACT

Access control is an important building block in many distributed applications, and several solutions, both centralised and distributed, have been proposed and used for such applications. Certificates are particularly well suited to distributed systems and have been used in several ways.

In this paper, we survey the certificate landscape from 2000 onwards. Our emphasis is on authorisation certificates and SPKI in particular.

Introduction

Valuable resources, such as home or credit accounts, require protection so that only authorised users can access them. To this end, various access control solutions are used for protection. For instance, the locks on our front doors prevent unauthorised guests from entering our homes, and when shopping, our credit card account can only be used with the corresponding credit card. But the need for protection is not limited to just private property – another example is public transport systems: without any control, a number of passengers would likely forgo buying a ticket.

In this survey, we examine the problem of protecting networked limited resources, and look at the different roles digital certificates play in access control from a life-cycle model perspective. We focus in particular on authorisation certificates and use one proposed solution, Simple Public Key Infrastructure (SPKI) authorisation certificates, to look at the solutions in more detail.

The rest of this paper is organised as follows. Section Access control in distributed applications provides a historical perspective on access control and explains the particularities of access control in distributed systems. Section Usage examples focuses on two application examples of access control and Section Phases of access control provides a life-cycle model for access control to help highlight the requirements for a distributed access control solution. Section Certificate based access control technologies discusses how certificates are used in access control and introduces SPKI certificates. Section Life-cycle view of access control technologies then provides a survey of the existing literature within the context of the life-cycle model. Finally, Section Discussion and future work presents our conclusions.

* Corresponding author.
E-mail address: Yki.Kortesniemi@aalto.fi (Y. Kortesniemi).

http://dx.doi.org/10.1016/j.cose.2014.03.013
0167-4048/© 2014 Elsevier Ltd. All rights reserved.

Access control in distributed applications

When discussing access control, we usually distinguish between *models and policies*, which decide who should and should not get access, and *mechanisms*, which implement and enforce the chosen model and policies. In this paper, the focus is on the mechanisms used for controlling access, all of which can be used to implement a variety of models and policies. Hence, we will not go into the different models and policies (for more details, see e.g. (Amoroso, 1994)).

In access control, we can identify three main roles: Resource, Administrator and User:

Resource is a valuable asset that needs to be protected. However, since an asset is often an inanimate object unable to act as a party in access control, we also often use the term resource to refer to the access control mechanism representing the resource. This mechanism enforces the rules and grants or denies access.

Administrator is the party deciding who will have access to the resource.

User is the party utilising the right to access the resource.

In simple systems, the same person can have multiple roles. For instance, if the resource is the front door to one's home, the administrator and user are the same person. In larger systems, however, there can be multiple administrator roles and even additional parties to overcome technical and even theoretical limitations. These exist because the basic model simply, but falsely, assumes, for example, that communication links always exist between parties wanting to communicate, that all related devices always have sufficient storage capacity and processing power, and that all of the required information exists when we want to use it. The additional parties enable us to provide additional computational and storage capacity and to overcome some communication challenges. Then, we can use smaller devices and often live with network partitioning.

Basics of access control

For the access control mechanism to function correctly, the resource must only accept decisions from the administrator. Otherwise, other malicious parties could issue access control decisions and incorrectly gain access to the resource or prevent authorised users from using the resource.

Further, for the administrator to *authorise* the user (i.e. grant access rights to the user), the user must have some identity to which the rights can be granted. Depending on the application, this identity does not have to be globally unique. Though the identity traditionally has been a user name, it can also be a pseudonym or even an ephemeral identity, such as a short-lived cryptographic identity. Also, in order for the user to be the only one to use this identity, there must be some way in which the user can authenticate itself as the proper User of the right. Here, it is important to make the distinction between the terms *identify* and *authenticate*. Identify means we recognise a party, but we do not use any means to ensure that they are the party they claim to be; with authentication, we require proof of their identity (SANS Institute, 2013).

A limited resource cannot tolerate unconstrained usage without problems. Therefore, even intended users typically have a quota, which defines how much they can use the resource. This quota can define, for instance, the total number of times that the resource can be used, such as a bus ticket that is valid for 10 trips. Alternatively, the quota could be a certain amount of money within a particular period of time, such as a credit card, which might have a quota of spending at most 5000 per month. To use such quotas, we need the chosen access control solution to support defining the usage limits and then enforce them.

Historically, digital access control (hereafter referred to as access control) began in the form of an access matrix, which listed all authorised entities and their rights in a table. As the number of users and the possible rights they could have grew, the table ended up growing very large – and yet, it was mostly empty, as each user only had a small subset of all possible rights. The solution was to split the table giving us two very different options: access control lists and capabilities (Amoroso, 1994).

Traditionally, the popular choice has been to base access control solutions on the concept of an Access Control List (ACL), where every resource is bundled with a list of authorised users. Typically, the list is located next to the resource, with all the relevant information in one place. In this solution, when the administrator wants to create a new right or change an existing one, she or he merely has to change the list. Furthermore, because the administrator controls the list, it is relatively easy to protect the integrity of the list, i.e. to make sure that the subject or any other outsider cannot change the contents of the list and thus create new or extended rights. Finally, because only the administrator is able to modify and issue lists, we can be sure that all information is authentic, i.e. that it comes from the correct, stated source. The downside of this solution is that the administrator cannot make any changes if he or she cannot access the list. Also, if there are multiple alternative resources, all of these would need access to a central ACL or their own copies of the ACL, which can result in a large amount of unnecessary information being passed around: for example, if we implemented credit cards using a replicated ACL in each shop, we would have to inform every shop in the world about every credit card being issued regardless of whether or not the user ever intended to visit the shop.

Capabilities reverse the concept, turning the centralised system into a distributed one. In a capability-based system, the users of the resource are given a ticket that proves they have the right to use the system. Credit cards are an example of this approach. Also, new capabilities can be created and given to the subject without any connection to the resources. So, the capability-based approach has some inherent advantages in large distributed systems compared to ACL: we can make access control decisions offline (as long as the information can be communicated before it is required) and we avoid flooding all resources with access control information they never need. The disadvantage is that we need additional mechanisms to revoke the rights since the administrator no longer controls the access right.

Finally, there are hybrid solutions, such as Kerberos (Steiner et al., 1988), which combine elements from both ends: