J. Lopez, R. Oppliger, and G. Pernul, "Authentication and Authorization Infrastructures (AAIs): A Comparative Survey", *Computers & Security*, vol. 23, pp. 578-590, 2004.
NICS Lab. Publications: https://www.nics.uma.es/publications

Authentication and Authorization Infrastructures (AAIs) for Identity Management: Comparative Analysis

Javier Lopez* Rolf Oppliger† Günther Pernul‡

Abstract

In this paper, we argue that traditional approaches for authorization and access control in computer systems (i.e., discretionary, mandatory, and role-based access controls) are not appropriate to address the requirements of networked and/or distributed systems, and that proper identity management requires infrastructural support. This support can be provided, for example, by Authentication and Authorization Infrastructures (AAIs). Against this background, we overview, analyze, discuss, and put into perspective some technological approaches that can be used to build and operate AAIs. More specifically, we address Microsoft .NET Passport and some related activities (e.g. the Liberty Alliance Project), Kerberos-based solutions, and AAIs that are based on digital certificates and Public Key Infrastructures (PKIs). We conclude with the insight that there is no single best approach for identity management, that every approach has specific advantages and disadvantages, and that a comprehensive AAI must combine various technologies and approaches .

*University of Malaga, Department of Computer Science, E.T.S. Ingenieria Informatica, Campus de Teatinos, E-29071 Malaga, Spain, Phone: +34 952 13 1327, Fax: +34 952 13 1397, E-mail: jlm@lcc.uma.es.

†eSECURITY Technologies Rolf Oppliger, Beethovenstrasse 10, CH-3073 Gümligen, Switzerland, Phone/Fax: +41 (0)79 654 8437, E-mail: rolf.oppliger@esecurity.ch.

‡University of Regensburg, Universitätsstrasse 31, D-93053 Regensburg, Germany, Phone: +49 (0) 941 943 2742, Fax: +49 (0) 941 943 2744, E-mail: guenther.pernul@wiwi.uni-regensburg.de.

1 Introduction

In a 1993 edition of *The New Yorker*, Peter Steiner published a cartoon[1] that showed a dog explaining to another dog the major advantage of the Internet, namely that "on the Internet, nobody knows you're a dog". In the following years, the cartoon was used by many security companies to promote *Public Key Infrastructures* (PKIs) for electronic commerce (e-commerce). The statement was made that an Internet merchant must know (the identity of) his customers, and that he would face a problem if he did not know that his customers were dogs. One may argue whether this statement actually hits the point. Would an Internet merchant really face a problem if he did not know that his customers were dogs? To answer this question, it is useful to look at the real world and to ask whether a merchant would face a problem if he did not know that his customers were dogs. In the real world we would probably say "yes". More interestingly, however, we would say "yes" not because the merchant dislikes dogs, but because the probability that the merchant gets money from a dog is negligible. As a result of risk analysis considerations, the merchant would probably refuse to serve the dog simply because he did not want to loose money. Consequently, there are (at least) two conclusions to draw:

1. Everything we do is subject to risk analysis (be it explicit or implicit).
2. The merchant may not care about the identity (or breed) of his customers if the risk of not getting paid is sufficiently small.

This line of argumentation leads to the insight that e-commerce requires authenticity in the foreground, and that *authorization* is very important from a commercial and practical point of view (note, for example, that a cash payment is only a special form of authorization). A merchant may be more interested in the authorization of his customers than in their authenticity (e.g., [Fei98]). This point has led to research and development activities that are collectively referred to as *trust management*. Trust management is a rather artificial term, and its use has been overblown in the past. In fact, one may argue that trust management is not particularly important and that all that matters is *risk management* [Gee98]:

> "Trust management is surely exciting, but like most exciting ideas it is unimportant. What is important is risk management, the sister, the dual of trust management. And because risk management makes money, it drives the security world from here on out".

To clarify this point, let us consider the situation in which a customer wants to order goods from an Internet merchant. In this situation, there are typically two questions the customer may ask:

1. Does he trust the merchant (to handle his order properly)?
2. Does he carry the risk of having the merchant not properly handling his order?

The first question is related to trust management, whereas the second question is related to risk management. In many situations, it is simpler and more efficient to elaborate on risks than it is to discuss trust (i.e., trust is difficult to address and even more difficult to quantify). In either case, it is important to note that trust and risks are not independent, and that the two things try to measure the same (or at least similar and closely related) things. For example, if we trust something, then

[1] The cartoon was published on page 61 of the July 5, 1993, issue of The New Yorker (Vol. 69, No. 20).

we usually mean that the risks involved using it are sufficiently small. Similarly, if we do not trust something or somebody, then we assume high risks.

Assuming that authorization is at least equally important than authentication, we may want to extend the scope of a security solution related to authentication (e.g., Kerberos or PKI) to simultaneously address authorization. In fact, there is an increasingly large body of research and development that elaborates on technologies to provide authentication and authorization services in networked and distributed systems. More recently, the term *identity management* was coined to refer to this area of research and development.

The aim of this work is to analyze and discuss some solutions that can be used to provide authentication or authorization services, and that fulfil the requirements of proper identity management. The rest of the paper is organized as follows: In Section 2, we elaborate on traditional approaches for authorization and access control. It is argued that these approaches stem from classical computer security, and that they are not particularly useful to address authorization and access control in networked and distributed environments. In Section 3, we analyze and discuss technologies to provide authentication or authorization services on the Internet; namely, .NET Passport, Kerberos-based technologies and PKI-based technologies. In Section 4 and 5, we introduce the idea of an *Authentication and Authorization Infrastructure* (AAI) and give evaluation criteria. In Section 6, we compare the technologies analyzed and discussed in Section 3. In Section 7, we conclude and give an outlook.

2 Traditional Approaches for Authorization and Access Control

The traditional approaches for authorization and access controls are overviewed and discussed in [Per95]. A major distinction has been made between discretionary and mandatory access control models:

- *Discretionary Access Control* (DAC) models are based on the concepts of a set of *objects* (i.e., the resources being protected), a set of *subjects* (i.e., the users or entities acting on behalf of the users), a set of *access rights* defining what kind of access a subject has to a certain object (e.g., read, write, execute, ...), and a set of *predicates* that may be used to represent constraints. DAC is based on two principles:

  1. Ownership of information;
  2. Delegation of rights.

  Ownership of information means that the creator of an object becomes its owner and ownership implies all permission with respect to this object. Delegation of rights means that the owner may also grant access to this item to other subjects. Most systems supporting DAC store access rules in an access control matrix (as introduced, for example, in [Lam71] and later refined in [GD72] and [HRU76]). For the reason of efficiency, an access control matrix is often stored column-wise (leading to object-specific access control lists) or row-wise (leading to subject-specific capability lists).

- *Mandatory Access Control* (MAC) models are more concerned with the flow of information within a system (rather than with the ownership of objects and the delegation of rights). More specifically, MAC requires that objects and subjects are assigned to certain *security classes* which are represented by

a *label*. The label for an object is called *classification*, whereas the label for a subject is called *clearance*. The classification represents the sensitivity of the labelled object, whereas the clearance represents the trustworthiness of a subject not to disclose sensitive information. A security label usually consists of two components:

1. A hierarchical set of *sensitivity levels* (e.g., top secret > secret > classified > unclassified);
2. A non hierarchical set of *categories*, representing classes of object types of the universe of discourse.

The sensitivity levels are totally ordered, but the set of categories are only partially ordered—thus, the set of classifications forms a lattice in a mathematical sense. In this lattice security class c_1 is comparable with and dominates c_2 if and only if the sensitivity level of c_1 is greater than or equal to that of c_2 and the categories in c1 contain those in c_2. Mandatory security grew out of the military environment where it is common practice to label information. This practice is also common in many companies and organizations where similar termed labels, like "confidential" or "company confidential", exist. The requirements are often based on the Bell-LaPadula security paradigm [BL76] and formalized by two rules:

1. Subject s is allowed to read object o only if $clearance(s) \geq classification(o)$. This rule protects information from unauthorized disclosure.
2. Subject s is allowed to write object o only if $clearance(s) \leq classification(o)$. This rule protects information from contamination or unauthorized modification by restricting the information flow from high to lower trusted subjects.

Mandatory security leads to multilevel systems because its content may appear differently to users with different clearances. This is because of two reasons:

1. Not all clearances may authorize all subjects to all data;
2. The support of MAC may lead to polyinstantiation of information. Polyinstantiation (i.e., multiple instances of a data item referring to a single fact of reality but differing by the classification label) for example is supported in multilevel secure databases.

Most operating systems in use today implement DACs, whereas MACs are mainly interested from a theoretical point of view. In either case, creating and maintaining proper access permissions in large organizations or in fast-changing environments is a dynamic, complex and very time-consuming task. Providing the resources necessary to carry out a specific function in an organization typically requires cooperation among different corporate human resources, computer and information systems, networks, and usually a broad collection of different departments is involved.

With conventional authorization and access control systems, introducing a new user to the organization would mean relating the user ID to every resource the user eventually needs to access in doing the job. The direct linking of users with permissions and resources is not only time-consuming, it invariably may lead to errors as user assignments change over the time, often resulting in users having permissions they should not have.

In *Role-Based Access Control* (RBAC) models, roles are the authorization subjects and are regarded as descriptions of organizational functions stating what has to be done regardless of who does it. Roles should imply only those permissions that are needed to fulfill the duties of a job carrying out the organizational function (implementing the principle of least privilege). RBAC does not permit users to be directly associated with permissions. With RBAC, permissions are authorized for roles and roles are authorized for users. In RBAC two different types of associations must be managed, associations between users and roles, and associations between roles and permissions. RBAC is a very dynamic but simple to administer model, for example, when a user's job position changes, only the user/role associations need to change. All the permissions of the involved roles remain the same. In 1996, Sandhu at al. [SCFY 96] introduced a framework of RBAC models, breaking down general RBAC into four conceptual models (i.e., RBAC0 to RBAC3). In 2000, NIST initiated an effort to establish a standard for RBAC which in 2002 was submitted for international standardization (see, for example, [FKC03]).

In addition to DAC, MAC, and RBAC, a few other models for authorization and access control have been proposed in the past. Examples are the Personal Knowledge Approach [BB88], the Clark and Wilson Model [CW87], and the Chinese Wall Policy [BN89]. None of these models has gained the amount of attention the above-mentioned models have achieved.

There is no doubt that solutions like DAC, MAC and RBAC are very useful in homogeneous environments. However, a networked and distributed environment, like the Internet, differs from a (stand-alone) computer system in the sense that it is typically not homogeneous. Different parties operate different systems with different operating systems and access control mechanisms. It is not possible to implement a homogeneous DAC, MAC, or RBAC. In fact, the sets of subjects, objects, access rights, and roles are not fixed and—even worse—are dynamically changing. Also, the different operators of the systems may use the same or similar labels to refer to different things. For example, what one operator considers to be secret, may be considered to be unclassified or top secret by some others. Consequently, the simplest approach would be to require a standardized notation for subjects, objects, access rights, etc. This approach, however, is considered to be impractical and alternative approaches are followed in this paper.

3 Analysis of Technologies for Authentication or Authorization Services on the Internet

There are some technologies available that can be used to provide authentication and authorization services on the Internet. Some of them are overviewed and briefly discussed next (more information can be found, for example, in [Opp03]).

3.1 Microsoft .NET Passport

As part of its .NET initiative, Microsoft has 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* (SSI[2]) service called

[2]Microsoft uses the term SSI to refer to the service that Microsoft .NET Passport provides. This is in contrast to the term *Single Sign-On* (SSO) that is otherwise used in the literature. It is not clear to what extent SSI differs from SSO in the terminology of Microsoft. Note, for example, that the term SSO is used in the documentation that describes Microsoft's Kerberos implementation in Windows 2000 and XP. In this article, we use the terms SSI and SSO synonymously and interchangeably.

Microsoft .NET Passport [Mic02][Mic03]. The service was initially released in 1999 and Microsoft claims that it is currently the most widely used service of its kind on the Internet.

Microsoft .NET Passport has a global name space, meaning that all users are uniquely identified with an e-mail address and all participating sites are uniquely identified with their DNS name. Also, Microsoft .NET Passport requires a shared secret for each *principal* or registered entity. For users the secret is a password, whereas for participating sites the secret is a cryptographic key. In either case, the secrets are centrally stored in a database. The central database also hosts the sign-in/sign-out pages, which participating .NET Passport sites can cobrand. The central database kept at Microsoft is a serious security and privacy risk. Who ever has access and controls the database also has access to all authentication means of all registered users. This is dangerous (to say the least).

When a user requests a resource from a participating site, his browser is redirected to Microsoft .NET Passport. Using an SSL/TLS connection between the browser and the .NET Passport server, the user credentials (i.e., username and password) are transmitted in cryptographically protected form. If the user has properly authenticated himself to the .NET Passport server, the browser gets a couple of cookies that encode relevant information. It is then redirected to the participating site and the request may be served accordingly. Using Microsoft .NET Passport, a user can easily move between participating sites without the need to remember a specific set of credentials for each of them.

There are two levels of security in addition to "standard sign-in":

- *Secure channel sign-in* where the browser and the participating site also employ an SSL/TLS connection to securely communicate with each other.
- *Strong credentials sign-in* where an additional four-digit secret key is used.

Note that participating .NET Passport sites rely on .NET Passport to authenticate users rather than hosting and maintaining their own authentication schemes. However, Microsoft .NET Passport does not authorize or deny a specific user's access to individual participating sites. Web sites that implement .NET Passport maintain control over permissions. As such, .NET Passport provides an authentication system or infrastructure, but does not provide a combined authentication and authorization service.

3.1.1 Liberty Alliance Project

More recently, some competitors of Microsoft launched a *Liberty Alliance Project*[3]. The project refers to an organization chartered "to create an open, federated, single sign-on identity solution for the digital economy via any device connected to the Internet". Membership is open to all commercial and non-commercial organizations. In July 2002, a first series of documents from the Liberty project was released.

According to these documents, the Liberty architecture distinguishes between identity providers and service providers. In short, identity providers provide identification and authentication services (similar to Microsoft .NET Passport), whereas service providers make use of these services to provide commercial services to users (similar to participating sites in the case of Microsoft .NET Passport). Contrary to Microsoft .NET Passport, the Liberty project does not assume a global name space. Instead, each participant may have several identities and identity federation and defederation are the basic building blocks in the Liberty project. In fact, in the terminology used in the Liberty project, single sign-on results from federated identities. Also contrary to Microsoft .NET Passport, cookies are not used to

[3] `www.projectliberty.org`

transfer information between identity providers and service providers. Instead, all information is transferred using HTTP redirects and URL encodings.

Although the Liberty Alliance Project looks promising, it's too early to tell whether it will be implemented and successfully deployed on the marketplace. In either case, the Liberty project is distributed and has some advantages with regard to scalability and interoperability.

3.2 Kerberos-based Technologies

Microsoft .NET Passport implements a simple and straightforward approach to provide password-based authentication. There is, however, another password-based authentication system, *Kerberos*, that is in widespread use[4]. This system can also be used to provide a starting-point for identity management.

The Kerberos authentication system was originally developed at MIT [SNS88,Sch94]. The first three versions of the system were used only internally, whereas version 4 was made publicly and freely available[5]. Some sites require functionality that Kerberos V4 does not provide, while others have a computing and networking environment or administrative procedures that differ from those at MIT. In addition, Bellovin and Merrit published a paper describing some shortcomings and limitations of Kerberos V4 in 1990 [BM90]. Against this background, work on Kerberos V5 commenced, also fueled by discussions with Kerberos V4 users and administrators about their experience with the Kerberos model in general, and the MIT reference implementation in particular. In 1993, Kerberos V5 was officially specified in RFC 1510 [KN93] and submitted to the Internet standards track. Again, MIT provided a publicly and freely reference implementation of Kerberos V5.

In short, Kerberos is based on authentication and key distribution protocols originally in [NS78, NS87] and modified to use timestamps [DS81]. In the Kerberos model and terminology, a domain is called a *realm*. The aim of Kerberos is to allow a client acting on behalf of a user to authenticate to a service (i.e., an application server) without having to send credentials (e.g., username and password) in the clear. Therefore, Kerberos implements a *ticketing system*. This basically means that principals request *tickets* from a trusted *Key Distribution Center* (KDC), and that these tickets are sent together with the service requests to authenticate the requesting principal. As illustrated in Figure 1, the KDC logically consists of an *Authentication Server* (AS) and a *Ticket Granting Server* (TGS). The aim of the AS is to issue *Ticket Granting Tickets* (TGTs), whereas the aim of the TGC is to issue service tickets. The service tickets are the ones a client must submit to authenticate itself to a server. Refer to [Opp96] or [Tun99] for details.

In its original form, Kerberos is an authentication and key distribution system that does not provide support for authorization and access control. There are, however, at least two initiatives that have tried to turn Kerberos into a full-fledged service for authentication and authorization.

beginfigure[htb]

[4]For example, Kerberos is used in the Windows 2000 and XP operating systems.

[5]Outside the United States and Canada, the eBones distribution of Kerberos V4 is heavily used and widely deployed. The eBones distribution is available at `http://www.pdc.kth.se/kth-krb/`.

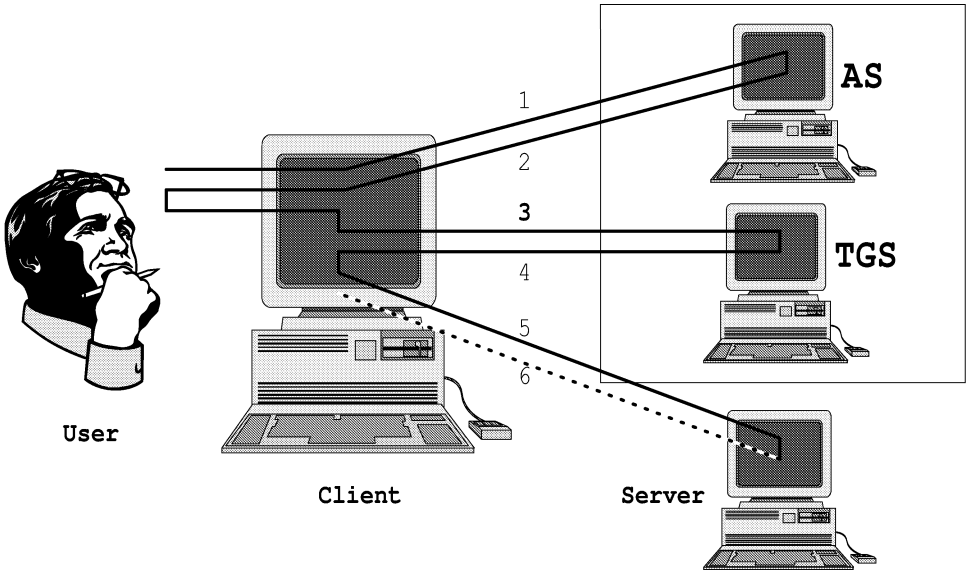

Figure 1 The Kerberos authentication protocol.

endfigure

3.2.1 SESAME

The *Secure European System for Applications in a Multi-vendor Environment* (SESAME) was an European research and development project aimed at developing a security infrastructure for distributed computing and networking environments [AV99]. It achieved this by including and combining an extended Kerberos authentication service and a *privilege attribute service* that can be used to provide authorization and corresponding access control services. The privilege attribute service, in turn, is provided by a *Privilege Attribute Server* (PAS) that issues digitally signed *Privilege Attribute Certificates* (PACs). In principle, a PAC consists of both the user's privileges and corresponding control information. The user's privileges are data such as the user's identity, role, organizational group, and security clearance, whereas the control information says where and when the PAC can be used and whether it can be delegated or not. Note that a PAC is conceptually similar to an attribute certificate (as further addressed below).

3.2.2 Kerberos version for Windows 2000

Microsoft implemented the Kerberos V5 authentication service with extensions for public key authentication[6] for its Windows 2000 operating system. The Kerberos KDC is integrated with other Windows 2000 security services running on the domain controller and uses the domain's active directory as its security account database.[7] In addition to the functionality specified in RFC 1510, Windows 2000 implements an authorization mechanism in the Kerberos system in a specific and unique way. When the Kerberos protocol is used for authentication, a list of *security identifiers* (SID) identifying a principal and the principal's group memberships is transported to the client in the authorization data field of a ticket (it is initialized for the ticket granting ticket and it is copied into each service ticket that is derived from it). It

[6]These extensions are specified by the IETF CAT WG under the acronym PKINT.

[7]For consistency, the Microsoft documentation uses the term "domain" instead of "realm". Furthermore, the distinction between an AS and a TGS is not made. Both components are collectively referred to as a KDC.

has been a hotly debated question in the security community whether the Kerberos version for Windows 2000 confoms to RFC 1510.

3.3 PKI-based Technologies

Microsoft .NET Passport and Kerberos both depend on user-selected passwords. This basically means that the overall security of the resulting systems are bounded by the security of the passwords. Unfortunately, all statistical investigations show that passwords selected by users have poor security properties (meaning, for example, that they can be guessed easily). Consequently, from a security point of view it is interesting to look into technologies that do not depend on users selecting "good" secrets (for any meaningful definition of "good") and use computer-generated secrets instead.

One such technology is public key cryptography (as originally proposed in [DH76]), and the use of digital certificates and PKIs [HFPS99, AL99, NDJB01]. In fact, digital certificates and PKIs can be used to provide an authentication infrastructure. Combined with some complementary technologies (e.g., attribute certificates), they can also be used as a starting-point to provide an authorization infrastructure.

3.3.1 ITU-T PMIs

An X.509v3 public key certificate can convey authorization information about its owner. The information can, for example, be encoded in one of the X.509v3 standard or extension fields. Note, however, that there are at least two reasons why caution should be taken in using X.509v3 public key certificates for conveying authorization information:

1. The authority that is most appropriate for verifying the identity of a person associated with a public key, i.e., a *Certification Authority* (CA), may not be appropriate for certifying the corresponding authorization information. For example, in a company the corporate security or human resources departments may be the appropriate authorities for verifying the identities of persons holding public keys, whereas the corporate finance office may be the appropriate authority for certifying permissions to sign on behalf of the company.

2. The dynamics of the two types of certificates may be fundamentally different. For example, the persons authorized to perform a particular function in a company may vary monthly, weekly, or even daily. Contrary to that, public key certificates are typically designed to be valid for a much longer period of time (e.g., 1 or 2 years). If it becomes necessary to revoke and reissue public key certificates frequently because of changing authorizations (that are encoded into the public key certificates extensions), this may have a severe impact on the performance characteristics of the resulting certificate management scheme.

Recognizing that public key certificates are not always the best vehicle to carry authorization information, the U.S. American National Standards Institute (ANSI) X9 committee developed an alternative approach known as *attribute certificates*. Meanwhile, this approach has been incorporated into both the ANSI X9.57 standard and the X.509-related standards and recommendations of ITU-T, ISO/IEC, and IETF.

According to RFC 2828 [Shir00], an attribute certificate is "a digital certificate that binds a set of descriptive data items, other than a public key, either directly

to a subject name or to the identifier of another certificate that is a public-key certificate". The latest version of the X.509 ITU-T recommendation [ITU00] specifies the format of an attribute certificate (currently in version 1). This certificate is a separate data structure from the public key certificate of the subject. ITU proposes the binding of both certificates (figure 2).

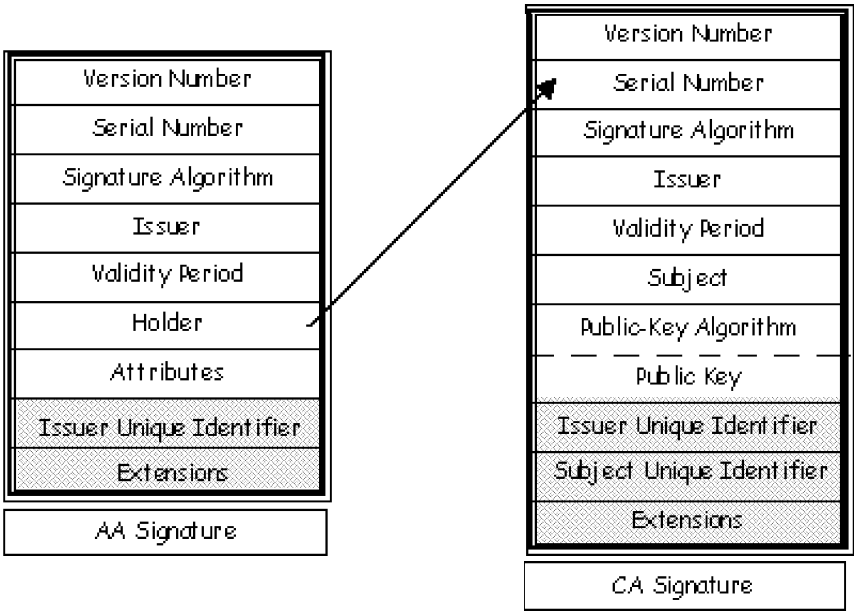

Figure 2 Binding attribute and identity certificates.

It can be seen that, similarly to public key certificate, an X.509 attribute certificate contains one field of information concerning the user. However, in this case the name of the field is not *subject* but *holder*. This is precisely the field used for the binding to the public key certificate. This way of binding allows that one subject has multiple attribute certificates associated with each of its possible public key certificates.

According to ITU recommendation an attribute certificate may be issued by a different entity, the *Attribute Authority* (AA), than the entity that issued the associated public key certificate (i.e., the CA). Thus, the attributes of a final user are digitally signed and its certificate issued by an AA, whose attributes are in turn signed and certificate issued by another AA. Chains of attribute certificates can be built recursively. In fact, the recommendation defines a framework that provides a foundation upon which a *Privilege Management Infrastructure* (PMI) is built.

A PMI contains a multiplicity of authorization relationships among AAs and final users, where the *Source of Authorization* (SOA) is the authority that a privilege verifier trusts as the ultimate authority to assign a set of privileges. Revocation procedures are also considered by defining the concept of *Attribute Certificate Revocation Lists* (ACRLs) which are handled in the same way as for CRLs published by CAs .

In spite of the fact that public key and attribute certificates are logically linked to each other, both types of certificates can be managed in independent infrastructures

(i.e., PKI and PMI) [DLMO02]. The mutual independence of the two infrastructures is also valid when considering other ways of describing the holder of the attribute certificate. In spite of using the serial number for the identity certificate it is possible to bind the attribute certificate to any object by using the hash value of that object. For instance, the hash value of the public key, or the hash value of the identity certificate itself can be used. The infrastructures are absolutely separated when considering the situation in which some other authentication method different from that one based on identity certificates is used. In these cases, a PKI is not even used. Hence, the name of the subject is a good option to describe the holder of the attribute certificate.

It is important to note that the use of PMIs result very flexible because ITU-T defines PMI models for different environments. Besides the general privilege management model, ITU defines:

1. *Control model*: Describes the techniques that enable the privilege verifier to control access to the object method by the privilege asserter, in accordance with the attribute certificate and the privilege policy.

2. *Roles model*: Individuals are issued *role assignment certificates* that assign one or more roles to them through the role attribute contained in the certificate. Specific privileges are assigned to a role name through *role specification certificates*, rather than to individual privilege holders through attribute certificates.

3. *Delegation model*: When delegation is used, the privilege verifier trusts the SOA to delegate a set of privileges to holders, some of which may further delegate some or all of those privileges to other holders.

3.3.2 SDSI/SPKI

The existence and usefulness of a globally unique namespace, such as the ones used in Microsoft .NET Passport and ITU-T X.509, has been challenged in the literature (e.g., [Ell96]). Most importantly, the *Simple Distributed Security Infrastructure* (SDSI) architecture and initiative [RL96] have evolved from the argument that a globally unique namespace is not appropriate for the global Internet, and that logically linked local namespaces provide a simpler and more realistic model [Aba97]. As such, work on SDSI also inspired the establishment of a *Simple Public Key Infrastructure* (SPKI) WG within the IETF. The WG was tasked with producing a certificate infrastructure and operating procedure to meet the needs of the Internet community for trust management in as easy, simple, and extensible way as possible.

Against this background, the IETF SPKI WG addressed the issue of <name,key>-bindings and realized that those certificates are of limited use for trust management because, in their opinion, a person's name is not of security interest. On the contrary, the WG believed that what a verifier of a certificate needs to know is whether a user is granted to perform certain operation [Ell99]. Thus, SPKI was initially only concerned of certifying bindings of public keys (which identify the keyholder) and attributes. However, the merge with SDSI has provided binding between names and public keys as well.

As stated above, the main purpose of an SPKI certificate is to authorize some action, give permission, grant a capability, etc. to or for a keyholder. The authorization scheme relies on sets of permission, and defines intersection operations on those sets. Principals are enabled to delegate subsets of their permissions to other principals, and can limit if it is allowed to further delegate (propagate) those permissions.

SPKI assumes that certificates are distributed directly by the keyholder to the verifier because certificates may carry sensitive information. This is also the reason why each SPKI certificate should carry the minimum information necessary to get permission. Another goal is that certificates can be used in very constrained environments, such as smart cards or PDAs, what confirms that they should be as simple as possible. SPKI uses *S-expressions* as the standard format for certificates [EFLRTY99]. A S-expression 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. SPKI also defines a canonical form for S-expressions.

A mechanism for deriving authorization decisions from a mixture of certificate types has been developed. In fact, certificates come in three categories: *ID* (mapping <name,key>), *Attribute* (mapping <authorization,name>), and *Authorization* (mapping <authorization,key>). In this way, when a principal wants to perform some action, generates a *tag* specifying the action that wants to perform. The tag is then embedded in a signed certificate. The result of this operation is the creation of a *claim* that is presented together with any evidence to the verifier. The verifier intersects the claim with the evidence and, using its own access control list, decides if the authorization is granted.

SPKI/SDSI does not claim to enforce one key per name. Therefore, a named group can be defined by issuing multiple (name,key) certificates with the same name – one for each group member.

4 Introducing the Notion of an AAI

In previous sections we have described and analyzed traditional approaches for authorization and access control in computer systems (i.e., discretionary, mandatory, and role-based access controls), and technologies that can be used to provide authentication and authorization services on the Internet. Generally speaking we can argue that all of those solutions concentrate either on authentication or authorization. Therefore, they fail to be complete because what we envision as necessary is to extend the scope of security solutions by providing an integrated authentication-and-authorization service for communicating peers; that is, to create an *Authentication and Authorization Infrastructure* (AAI).

Using an AAI, a user typically registers only once in his or her home domain. When the user requests a resource, he or she should always be authenticated by his or her home domain, and the request should be forwarded to the destination server complemented with some additional information (provided by the user's home domain authentication server). Consequently, the challenge of an AAI is to provide an inter-domain authentication and authorization service.

The situation is comparable to roaming agreements in GSM networks, where subscribers are registered by their *Home Location Registers* (HLRs) but can also be authenticated and authorized by *Visited Location Registers* (VLRs). The important point to note is that VLRs do not need to register the users for themselves; instead they trust the registration process and the credentials provided by the HLRs. They only focus on local authorization and access control decisions.

This idea may be adapted for AAIs too. Typically, every domain will operate one (or several) authentication and authorization server(s). The registration of the principal will take place in his or her *home domain*, getting credentials that can be used to authenticate and authorize to principals in other domains (visited domains from the visiting principal's point of view). In either case, a principal will need to store its credentials either locally or remotely. In the second case, a secure credentials download protocol is required.

5 Evaluation Criteria

In order to evaluate how existing technological approaches for AAIs suit into the needs, it is necessary to clearly establish the requirements of the latter. Therefore, in this subsection we identify and briefly explain the features that we conceive as fundamental for an identity management that is both efficient and effective. Obviously, these features will be used as the evaluation criteria for existing approaches. Comparison of approaches is done in next section.

Security: An AAI should be secure in the sense that it is computationally infeasible for a malicious principal to spoof the (registered) identity and/or illegitimately use the credentials of another principal. This also implies that credentials must either be securely stored locally, or securely downloaded from a credentials repository.

Efficiency: An AAI should be efficient in the sense that the provision of authentication and authorization services should not require too much resources (both for computation and communication), while enforcing clear and consistent policies.

Scalability: An AAI should be scalable in the sense that it can be used for a potentially very large community of users; that is, it must be able to manage credentials for a large distributed work force. Scalability is particularly important for the provision of inter-domain authentication and authorization services.

Interoperability: An AAI should be able to handle different types of credentials (e.g., usernames and passwords, public key certificates, Kerberos tickets, Microsoft Passports, ...). The credentials, in turn, should be as simple as possible and not include proprietary information.

Delegation: An AAI should provide the ability to delegate authorizations from one user to another without bothering the owner of the resource(s) involved. It should be necessary to distinguish between simple permissions (e.g., to read some file) and permissions to delegate that permission further. In this last case, two issues arise: the capacity to limit depth of delegation, and the question of separating delegators from those who can exercise the delegated permission.

Revocation: An AAI should allow authorities, system managers, permission managers, etc. to revoke privileges of users. This procedure should be easy. Moreover, revocation information should be available for the rest of users in the system or domain in the smallest slice of time as possible. Note, however, that revocation of privileges in the case transitive delegations are possible is a very challenging issue (that is not further addressed in this paper).

Privacy: An AAI should provide mechanisms that allow that privileges, roles, clearances, etc. of users are only known to verifiers. That is, the user or the manager should have the ability to protect such information, from eavesdroppers or even other legitimate users in the system, when it travels to the verifier. Furthermore, in case anonymous or pseudonymous access is desired, the AAI may act as an anonymizing or pseudonymizing service.

Mobility: An AAI should provide an essential feature for Internet scenarios: the means to transport authentication and authorization information to decentralized applications. That is, the information must become "mobile", which is essential in e-commerce applications.

Suitability for mobile computing : This is a feature that evolves form previous one. Here, we mean that the AAI should provide some mechanisms that allow to use authorization information not only on Internet connected desktop computers (what originates the "mobility" feature) but also to use such information in devices of limited capacity of storage and processing, like smartcards, PDAs, etc.

In either case, it is important to consider the naming procedures used by the authorization scheme. The reason is that local names schemes may constraint essential features like scalability. On the other hand, global unique names procedures may be unrealistic and not efficient.

6 Comparison

In this section we discuss how the approaches introduced in Section 3 meet the evaluation criteria itemized in the previous section, or what is to say, how these approaches fulfil the AAI requirements for identity management. The situation is overviewed in Table 1 ("+" refers to an advantage, whereas "-" refers to a disadvantage). The table can be seen as the outcome of the analysis of technologies performed in section 3.

Table 1: Evaluation of the technological approaches for AAIs

Evaluation Criteria	Microsoft .NET Passport	Kerberos-based AAIs	PKI-based AAIs
Security	+	+	++
Efficiency	+	+	++
Scalability	-	-	++
Interoperability	-	+	++
Delegation	-	++	++
Revocation	++	++	- -
Privacy	–	-	+
Mobility	+	-	-
Mobile computing	+	-	-

Referring to the evaluation criteria itemized in Section 5, one can argue that Microsoft .NET Passport is reasonably secure (at least if "secure channel sign-in" or "strong credentials sign-in" are used), and that it runs efficiently (it uses standard Web technologies). Due to the fact that Microsoft .NET Passport is a proprietary and centralized service, it has poor scalability and interoperability properties. Furthermore, as it does not address authorization, delegation is not an issue. Its online nature makes revocation automatically handled. However, due to its centralized nature, Microsoft .NET Passport has very bad privacy characteristics (this must not necessarily be the case for the Liberty Alliance Project). Finally, its extensive use of Web technologies results in Microsoft .NET Passport having good properties regarding mobility and suitability for mobile computing.

Similar to Microsoft .NET Passport, Kerberos-based AAIs have sufficiently good security and efficiency properties. Scalability is poor, but interoperability is good. This is because the Kerberos protocol is standardized and supported by many platforms. In the Kerberos model, there are specific tickets that can be used for delegation. Consequently, delegation can be supported. The same is true for revocation. Again, Kerberos requires a centralized server that can be used to immediately revoke

tickets. The centralized server, however, also has a privacy disadvantage (although this disadvantage is smaller than with Microsoft .NET Passport because each realm runs its own database). Last but not least, the fact that applications must be Kerberized to make use of a Kerberos-based AAIs is disadvantageous with regard to mobility and suitability for mobile computing.

Contrary to Microsoft .NET Passport and Kerberos-based AAIs, PKI-based AAIs can be designed in a way that provide very good security, efficiency, scalability, and interoperability, which are features directly inherited from the design of PKIs. Furthermore, having attribute certificates in mind, delegation can be provided in an a appropriate way, as deduced from the definition of the PMI delegation model in Section 3. The Achilles heel of the technological approach, however, is revocation. In fact, providing support for certificate revocation requires the existence of a trusted online server. PKI-based AAIs can potentially be designed to provide more privacy than Microsoft .NET Passport and Kerberos-based AAIs. The degree of privacy, however, depends on the design of the AAI. In fact, it is possible to encrypt attributes when an attribute certificate is carried in clear within an application protocol, or when it contains some sensitive information [FaHo02]. Additionally, there are, for example, some minimum-disclosure certificate technologies that can be used to maximize privacy (from a user's point of view). Regarding mobility, PKI-based AAIs are well suited for the transport of authentication and authorization information in decentralized scenarios. On the other hand, the length of standard digital certificates, as well as the regular time consumed for their processing, does not recommend their use in mobile computing.

7 Conclusions and Outlook

In this paper, we argued that traditional approaches for authorization and access control in computer systems are not appropriate to address the requirements of networked and/or distributed systems, that proper identity management requires some infrastructural support, and that this support can be provided, for example, by AAIs.

Against this background, we overviewed, analyzed, discussed, and put into perspective some technological approaches (.NET Passport, and Kerberos-based and PKI-based technologies) to build and operate AAIs. The approaches have specific advantages and disadvantages (see Table 1). This makes it difficult to compare them with each another. Furthermore, it is reasonable to expect the world to behave in a rather heterogeneous way.

Consequently, the border gateways of a domain will have the task to dealing with user credentials and to use some heuristics regarding the validation of these credentials. There is much room for further research and development in this area.

References

[**Aba97**] M. Abadi, "On SDSI's Linked Local Name Spaces," *Proceedings of 10th IEEE Computer Security Foundations Workshop*, June 1997, pp. 98–108.

[**AL99**] C. Adams and S. Lloyd, *Understanding Public-Key Infrastructure: Concepts, Standards and Deployment Considerations*, New Riders, 1999

[**AV99**] P. Ashley and M. Vandenwauver, *Practical Intranet Security—Overview of the State of the Art and Available Technologies*, Kluwer Academic Publishers, Norwell, MA, 1999.

[**BB88**] J. Biskup and H.H. Brüggemann, The Personal Model of Data: Towards a Privacy-Oriented Information System, *Computers and Security*, Vol. 7, 1988.

[**BL76**] D.E. Bell and L.J. LaPadula, *Secure Computer System: Unified Exposition and Multics Interpretation.* Technical Report MTR-2997. MITRE Corp. Bedford, Mass, 1976.

[**BM90**] S.M. Bellovin and M. Merritt, "Limitations of the Kerberos Authentication System," *ACM Computer Communication Review*, Vol. 20, 1990, pp. 119–132.

[**BN89**] D.F.C. Brewer and M.J. Nash, "The Chinese Wall Security Policy," *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 1989

[**CW87**] D.D. Clark and D.R. Wilson, "A Comparison of Commercial and Military Computer Security Policies," *Proceedings of the IEEE Symposium on Research in Security and Privacy*, 1987.

[**DS81**] D.E. Denning and G. Sacco, "Timestamps in Key Distribution Protocols," *Communications of the ACM*, Vol. 24, 1981, pp. 533–536.

[**DH76**] W. Diffie, M. Hellman, "New Directions in Cryptography". *IEEE Transactions on Information Theory*, IT-22, n. 6. 1976, pp. 644-654.

[**DLMO02**] E. Dawson, J. Lopez, J. A. Montenegro, E. Okamoto, "A New Design of Privilege Management Infrastructure for Organizations Using Outsourced PKI", Fifth International Conference on Information Security (ISC'02), pp.136–149, Lecture Notes in Computer Science v.2433, Springer-Verlag, September 2002.

[**Ell96**] C. Ellison, "Establishing Identity Without Certification Authorities," *Proceedings of USENIX Security Symposium*, 1996.

[**Ell99**] C. Ellison, "SPKI Requirements", *Request for Comments 2692*, Network Working Group, Internet Engineering Task Force, September 1999.

[**EFLRTY99**] C. Ellison, B. Frantz, B. Lampson, R. Rivest, B. Thomas, T. Ylonen, "SPKI Certificate Theory", *Request for Comments 2693*, Network Working Group, Internet Engineering Task Force, September 1999.

[**FaHo02**] S. Farrell, R. Housley, "An Internet Attribute Certificate Profile for Authorization" *Request for Comments 3281*, Network Working Group, Internet Engineering Task Force, April 2002.

[**Fei98**] J. Feigenbaum, "Towards an Infrastructure for Authorization," Position Paper, *Proceedings of USENIX Workshop on Electronic Commerce—Invited Talks Supplement*, 1998, pp. 15–19.

[**FKC03**] D. Ferraiolo, R. Kuhn, and R. Chandramouli, *Role-based Access Controls*, Artech House Publishers, Norwood, MA, to appear in 2003.

[**GD72**] G.S. Graham and P.J. Denning, "Protection Principles and Practices," *Proceedings of the AFIPS Spring Joint Computer Conference*, 1972, pp. 417–429.

[**Gee98**] D. Geer, "Risk Management Is Where the Money Is" Digital Commerce Society of Boston, November 1998, `http://catless.ncl.ac.uk/Risks/20.06.html#subj1`.

[**HFPS99**] R. Housley,W. Ford, W. Polk, D. Solo, "Internet X.509 Public Key Infrastructure Certificate and CRL Profile", *Request for Comments 2459*, Network Working Group, Internet Engineering Task Force, January 1999.

[**HRU76**] M.A. Harrison, W.L. Ruzo, and J.D. Ullman, "Protection in operating systems," *Communications of the ACM*, Vol. 19, No. 8, August 1976.

[**HWL00**] J. Hwang, K. Wu, and D. Liu, "Access Control with Role Attribute Certificates," *Computer Standards and Interfaces*, Vol. 22, March 2000, pp. 43–53.

[**ITU97**] ITU-T Recommendation X.509, "Information Technology. Open systems interconnection. The Directory: Authentication Framework," June 1997.

[**ITU00**] ITU-T Recommendation X.509, "Information Technology. Open systems interconnection. The Directory: Public-key and attribute certificate frameworks," March 2000.

[**KN93**] J. Kohl, C. Neuman, "The Kerberos Network Authentication Service (V5)", *Request for Comments 1510*, Network Working Group, Internet Engineering Task Force, September 1993.

[**KR00**] D.P. Kormann, and A.D. Rubin, "Risks of the Passport Single Signon Protocol," *Computer Networks*, Vol. 33, 2000, pp. 51–58.

[**Lam71**] B.W. Lampson, "Protection. Proc. Princton Conf. on Information and Systems Sciences. March 1971.

[**Mic02**] Microsoft, ".NET Passport: Balanced Authentication Solutions", December 2002.

[**Mic03**] Microsoft, "Microsoft .NET Passport Review Guide", March 2003.

[**NDJB01**] A. Nash, W. Duane, C. Joseph, D. Brink, "PKI: Implementing and Managing E-Security", McGraw-Hill, 2001.

[**NS78**] R.M. Needham and M.D. Schroeder, "Using Encryption for Authentication in Large Networks of Computers," *Communications of the ACM*, Vol. 21, December 1978, pp. 993–999.

[**NS87**] R.M. Needham and M.D. Schroeder, "Authentication Revisited," *ACM Operating Systems Review*, Vol. 21, 1987, p. 7.

[**Opp96**] R. Oppliger, *Authentication Systems for Secure Networks*, Artech House, Norwood, MA, 1996.

[**Opp03**] R. Oppliger, *Security Technologies for the World Wide Web, Second Edition*, Artech House Publishers, Norwood, MA, 2002.

[**OPS00**] R. Oppliger, G. Pernul, and Ch. Strauss, "Using Attribute Certificates to Implement Role-based Authorization and Access Control," *Proceedings of the 4. Fachtagung Sicherheit in Informationssystemen (SIS 2000)*, October 2000, pp. 169–184.

[**Per95**] G. Pernul, "Information Systems Security: Scope, State-of-the-Art, and Evaluation of Techniques," *International Journal of Information Management*, Vol. 15, No. 3, pp. 242–256.

[**RL96**] R.L. Rivest and B. Lampson, "SDSI—A Simple Distributed Security Infrastructure," April 1996.

[**SCFY96**] R.S. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman, "Role-Based Access Control Models," *IEEE Computer*, Vol. 29, Nr. 2, pp. 38–47.

[**Sch94**] J.I. Schiller, "Secure Distributed Computing," *Scientific American*, November 1994, pp. 72–76.

[**Shir00**] R. Shirey, "Internet Security Glossary", *Request for Comments 2828*, Network Working Group, Internet Engineering Task Force, May 2000.

[**SNS88**] J.G. Steiner, B.C. Neuman, and J.I. Schiller, "Kerberos: An Authentication Service for Open Network Systems," Position Paper, *Proceedings of the USENIX UNIX Security Symposium*, August 1988.

[**Tun99**] B. Tung, *Kerberos: A Network Authentication System*, Addison-Wesley, Reading, MA, 1999.