

Refined Access Control in a Distributed Environment

Erik Boström

LITH-ISY-EX-3257-2002
2002-01-15

Refined Access Control in a Distributed Environment


Master's thesis in Information Theory
Linköping Institute of Technology

Erik Boström

LiTH-ISY-EX-3257-2002

Supervisors: Johan Otterström, Peter de Laval, Sectra Communications
Examiner: Viiveke Fåk

Linköping 15 January 2002

	Avdelning, Institution Division, Department Institutionen för Systemteknik 581 83 LINKÖPING	Datum Date 2002-01-15								
Språk Language Svenska/Swedish X Engelska/English	Rapporttyp Report category Licentiatavhandling X Examensarbete C-uppsats D-uppsats Övrig rapport	ISBN ISRN LITH-ISK-EX-3257-2002 <table border="0"> <tr> <td>Serietitel och serienummer</td> <td>ISSN</td> </tr> <tr> <td>Title of series, numbering</td> <td></td> </tr> </table>	Serietitel och serienummer	ISSN	Title of series, numbering					
Serietitel och serienummer	ISSN									
Title of series, numbering										
URL för elektronisk version http://www.ep.liu.se/exjobb/isy/2002/3257/										
<table border="0"> <tr> <td>Titel</td> <td>Finkornig åtkomstkontroll i en distribuerad miljö</td> </tr> <tr> <td>Title</td> <td>Refined Access Control in a Distributed Environment</td> </tr> <tr> <td>Författare</td> <td>Erik Boström</td> </tr> <tr> <td>Author</td> <td></td> </tr> </table>			Titel	Finkornig åtkomstkontroll i en distribuerad miljö	Title	Refined Access Control in a Distributed Environment	Författare	Erik Boström	Author	
Titel	Finkornig åtkomstkontroll i en distribuerad miljö									
Title	Refined Access Control in a Distributed Environment									
Författare	Erik Boström									
Author										
Sammanfattning Abstract <p>In the area of computer network security, standardization work has been conducted for several years. However, the sub area of access control and authorization has so far been left out of major standardizing.</p> <p>This thesis explores the ongoing standardization for access control and authorization. In addition, areas and techniques supporting access control are investigated. Access control in its basic forms is described to point out the building blocks that always have to be considered when an access policy is formulated. For readers previously unfamiliar with network security a number of basic concepts are presented. An overview of access control in public networks introduces new conditions and points out standards related to access control. None of the found standards fulfills all of our requirements at current date. The overview includes a comparison between competing products, which meet most of the stated conditions.</p> <p>In parallel with this report a prototype was developed. The purpose of the prototype was to depict how access control could be administered and to show the critical steps in formulating an access policy.</p>										
Nyckelord Keyword access control, PKI, VPN, X.509, RBAC, role-based access control, computer security, cryptography										

Abstract

In the area of computer network security, standardization work has been conducted for several years. However, the sub area of access control and authorization has so far been left out of major standardizing.

This thesis explores the ongoing standardization for access control and authorization. In addition, areas and techniques supporting access control are investigated.

Access control in its basic forms is described to point out the building blocks that always have to be considered when an access policy is formulated.

For readers previously unfamiliar with network security a number of basic concepts are presented.

An overview of access control in public networks introduces new conditions and points out standards related to access control. None of the found standards fulfills all of our requirements at current date. The overview includes a comparison between competing products, which meet most of the stated conditions.

In parallel with this report a prototype was developed. The purpose of the prototype was to depict how access control could be administered and to show the critical steps in formulating an access policy.

Contents

1	INTRODUCTION.....	1
1.1	ASSIGNMENT	1
1.2	READING INSTRUCTIONS	3
1.3	SECTRA.....	4
2	ACCESS CONTROL IN PRIVATE NETWORKS	5
2.1	TERMINOLOGY	5
2.2	ACCESS OPERATIONS	7
2.3	OWNERSHIP	8
2.4	ACCESS CONTROL STRUCTURES	8
2.5	INTERMEDIATE CONTROLS.....	11
2.6	OPERATING SYSTEMS.....	17
2.7	SUMMARY	21
3	SECURITY CONCEPTS IN PUBLIC NETWORKS.....	23
3.1	EXAMPLE OF SECURITY ISSUES.....	24
3.2	CRYPTOGRAPHY	25
3.3	FIREWALLS.....	28
3.4	PUBLIC-KEY INFRASTRUCTURE (PKI)	30
3.5	VIRTUAL PRIVATE NETWORKS.....	33
3.6	SUMMARY	35
4	ACCESS CONTROL IN PUBLIC NETWORKS	37
4.1	SECURITY MANAGEMENT CENTER (SMC).....	37
4.2	ACCESS CONDITIONS.....	38
4.3	STANDARDIZATION WORK.....	40
4.4	PRODUCTS	44
4.5	SUMMARY	46
5	PROTOTYPE.....	47
5.1	BACKGROUND	47
5.2	DESIGN	48
5.3	EVALUATION	50
5.4	SUMMARY	51
6	CONCLUSION	53
6.1	GENERAL CONCLUSIONS	53
6.2	CONCLUSIONS CONCERNING THE SMC IN LWK.....	54
6.3	A LOOK INTO THE FUTURE.....	54
7	BIBLIOGRAPHY	57
APPENDIX A	ACRONYMS.....	61
APPENDIX B	GLOSSARY	63

APPENDIX C	SECURITY MANAGEMENT CENTER (SMC)	71
C.1	PURPOSE	71
C.2	ENVIRONMENT OF THE SMC	71
C.3	DESIGN	71
C.4	INTERNAL SYSTEM INTERFACES	73
C.5	EXTERNAL SYSTEM INTERFACES	75
APPENDIX D	STANDARDIZATION	77
D.1	STANDARDIZATION ORGANIZATIONS	77
D.2	STANDARDS AND PRODUCTS	81
APPENDIX E	EXTERNAL PRODUCTS	86
E.1	DCE BY OPEN GROUP	86
E.2	SELECTACCESS BY BALTIMORE TECHNOLOGIES	90
E.3	VPN-1 BY CHECK POINT	93

List of Figures

FIGURE 1: <i>THE FUNDAMENTAL MODEL OF ACCESS CONTROL</i>	5
FIGURE 2: <i>PROTECTION RINGS</i>	7
FIGURE 3: <i>GROUPS AS AN INTERMEDIATE LAYER OF ACCESS CONTROL</i>	11
FIGURE 4: <i>ACCESS CONTROL WITH NEGATIVE PERMISSIONS</i>	12
FIGURE 5: <i>PRIVILEGES BETWEEN SUBJECTS AND OPERATIONS</i>	13
FIGURE 6: <i>MODEL OF ROLE-BASED ACCESS CONTROL (RBAC)</i>	14
FIGURE 7: <i>PUBLIC NETWORK SECURITY</i>	23
FIGURE 8: <i>MODEL OF ENCRYPTION</i>	25
FIGURE 9: <i>POSITION OF A FIREWALL IN A NETWORK</i>	29
FIGURE 10: <i>ARCHITECTURE OF PUBLIC KEY INFRASTRUCTURE (PKI)</i>	31
FIGURE 11: <i>COMPONENTS OF A VIRTUAL PRIVATE NETWORK (VPN)</i>	34
FIGURE 12: <i>ENVIRONMENT OF THE SMC</i>	37
FIGURE 13: <i>COMPONENTS OF A RULE</i>	48
FIGURE 14: <i>A TREE SAMPLE WITH NODES AND CORRESPONDING ID-NUMBERS</i>	48
FIGURE 15: <i>MANAGING SUBJECTS, OBJECTS, OPERATIONS, AND CONDITIONS</i>	49
FIGURE 16: <i>MANAGING POSITIVE AND NEGATIVE RULES AND A TEST TOOL</i>	50
FIGURE 17: <i>CROSSING POSITIVE AND NEGATIVE RULES</i>	51
FIGURE 18: <i>FOUR LAYERS OF SECURITY ENGINEERING</i>	69
FIGURE 19: <i>ENVIRONMENT OF THE SMC</i>	71
FIGURE 20: <i>OVERALL STRUCTURE OF THE SMC</i>	72
FIGURE 22: <i>EXAMPLE OF ACL-BASED AUTHORIZATION</i>	89
FIGURE 23: <i>COMPONENT ARCHITECTURE IN SELECTACCESS</i>	91

List of Tables

TABLE 1: <i>PROTECTION LEVELS IN A PROTECTION RING</i>	6
TABLE 2: <i>AN ACCESS CONTROL MATRIX</i>	9
TABLE 3: <i>EXAMPLE OF CAPABILITIES, SPECIFYING SUBJECTS' ACCESS RIGHTS</i>	9
TABLE 4: <i>ACCESS CONTROL LISTS (ACLs)</i>	10
TABLE 5: <i>KEY USAGE IN PUBLIC KEY CRYPTOGRAPHY</i>	28
TABLE 6: <i>THE OSI AND TCP/IP REFERENCE MODELS</i>	67

1 Introduction

Today various standardized protocols exist as well as interfaces concerning network security. They provide integrity control, confidentiality for data sent over an insecure channel, and authentication when a connection is about to be established. At the moment, standardization of access control and authorization is in progress. One organization involved in standardization for the Internet is IETF (Internet Engineering Task Force). Parts of their standards consider access control.

Access control in operating systems has been available since the seventies, for example in Multics [Organick 1972]. In contrast, access control in heterogeneous distributed systems is not as well established. A reason to this is that interoperability needs to be considered which is rather cumbersome to fulfill.

Interoperability is the key subject in our discussion because that is why we want to find a standard. If the interoperability did not matter, we could go for an existing product and be happy with that. But to be sure that our system is going to work together with other systems (i.e. be interoperable), a standard is needed. Of course, in that case we take for granted that the other systems are also using the same standard.

1.1 Assignment

Sectra has developed a security infrastructure named LWK (LAN/WAN Krypto). The infrastructure provides two secure functions; transport protection and data object protection. To support these functions there is a security server, SMC (Security Management Center), which among other things governs access control.

Today's LWK has the following properties:

- Session keys are used, thus the master keys are valid during long periods.
- The SMC constitutes a protected environment for the key generation.
- LWK takes care of basic access control.

Even so, some improvements are wanted in the current product, which are related to:

- Low granularity of the access control. Hence, a user is granted access to every resource on the server, or no resource at all. Desired is that a user could gain access to only a subset of the resources. This is what we call fine-grained access control.
- Authentication is carried out with symmetric keys. The problem here is that most of the available standards are using asymmetric keys. An adjustment from asymmetric to symmetric keys is perhaps needed.

1.1.1 Purpose

Our goal is to explore ongoing standardization work in the area of access control and authorization. That includes a study of the basics in access control and supporting security principles. A sub goal is to perform an inquiry into the opportunities of merging suitable standards with Sectra's future projects. One of the future projects will involve VPNs (Virtual Private Networks). Hence, especially access control in VPNs should be investigated.

Finally, a prototype is developed in parallel with this report. The purpose of the prototype is to depict how access control could be administered and to show the critical steps in formulating an access policy. Hopefully, the prototype will help to determine directions for future investigations of how access control should be deployed.

1.1.2 Limitations

The scope of the thesis is access control in distributed systems. The parts of computer security that are needed as a base for access control are treated. No restrictions are done on access permissions. Thus, the access conditions can be arbitrary.

However, only principles are considered and not much attention is paid to underlying details of implementation.

Two sub areas are investigated more deeply and the following questions depicts those areas:

- How should the access rules be defined and used?
- What is included in the concept of users? (individuals, groups or roles)

1.1.3 Methodology

The assignment has been solved in the following manner. To start with, I had to read a lot just to come to grips with this specific area. The briefing included reading available literature at Sectra, although Internet has been the main source of information. After one month I started to write on this report. Now and then I had meetings with my tutors to discuss the work progress.

When I had a general picture of access control, I started to analyze and made an appraisal of the information I had collected. This work was followed by stating problems with possible solutions and corresponding motivations.

The prototype involved four phases; specifying requirements, specifying design, implementation in Visual Basic and finally a test phase to evaluate the prototype.

The work included several iterations among the steps mentioned above, which is quite natural.

1.2 Reading Instructions

No prerequisites are necessary as most concepts used in this thesis are explained. However, having some basic knowledge of computer security and network protocols will help in understanding the contents.

Chapter 2: Access Control in Private Networks covers the building blocks of access control and deals with how access control is achieved in operating systems.

Chapter 3: Security Concepts in Public Networks gives an overview of common security concepts and techniques used in public networks. The chapter serves as a knowledge base for the next chapter.

Chapter 4: Access Control in Public Networks points out the new requirements on access control that are introduced in public networks. In addition, standards and products are examined in order to see how the new requirements are dealt with.

Chapter 5: Prototype describes requirements for the developed prototype as well as an evaluation of the latter.

Chapter 6: Conclusion states the results of the investigated standards and products and points out suggestions for future work.

Chapter 7: Bibliography

App A: Acronyms consists of the acronyms used in the report.

App B: Glossary is intended to give very concise information about terms and concepts used. The glossary is supposed to work as a small dictionary for the report.

App C: Security Management Center is an overview of the product SMC developed by Sectra.

App D: Standardization examines organizations that are involved in standardization of access control and related standards, together with the standards and products they have conducted.

App E: External Products describes three products that have implemented access control in a distributed environment.

1.3 Sectra

Sectra AB has its roots in Linköping Institute of Technology and is one of Sweden's fastest growing high-tech companies in the IT area. Since the mid 1980s, Sectra has conducted development and sales of high-technology medical IT and telecommunications products. Today, the business includes products in medical imaging and information systems, secure communication systems and wireless information systems.

Business operations are conducted in three companies, Sectra Imtec AB, Sectra Communications AB and Sectra Wireless Technologies AB, all three wholly owned subsidiaries of Sectra AB.

The place for my thesis work is Sectra Communications AB, which develops and supplies encrypted communication systems positioned in the high-end of the product range. Products include defense grade encryption, hardwired encryption kernels and other high-security features. Sectra Communications AB is the largest supplier of encryption products to the Swedish Armed Forces.

The Sectra main office is located in Linköping, Sweden, and is headquarter for the Sectra group and the three business areas. In total, Sectra AB has almost 200 employees and offices in six countries.

2 Access Control in Private Networks

This chapter starts with a description of common components in access control, which is heavily inspired by *Computer Security* [Gollmann 1999].

Before throwing ourselves into details of access control, consider the way computer systems have developed over the last few decades. First, there were single-user systems and in those systems, the most important security property was integrity. The different parts of a system needed to be separated so that they could not interfere (i.e. alter each others data). As the parts had to communicate some mediating mechanism was needed. This mechanism included access control.

Our next step is multi-user operating systems where the integrity problem is extended with keeping private user information secret. Hence, confidentiality is introduced since not only alteration has to be controlled, but also observation. These systems are called *private networks* in this report and are the topic of this chapter.

The last step of access control comes with the *public networks*. Their insecure communication channels distinguish them from private networks. In public networks confidentiality is extended because now the data needs protection during transmission and not only while it is stored. However, security in public networks is covered in the chapters 3 and 4.

2.1 Terminology

The basic building blocks of access control are an active *subject* accessing a passive *object* with some specific *operation* request, while a *reference monitor* grants or denies access. The reference monitor is in charge of the access permissions for each object. This fundamental model of access control is shown in Figure 1.

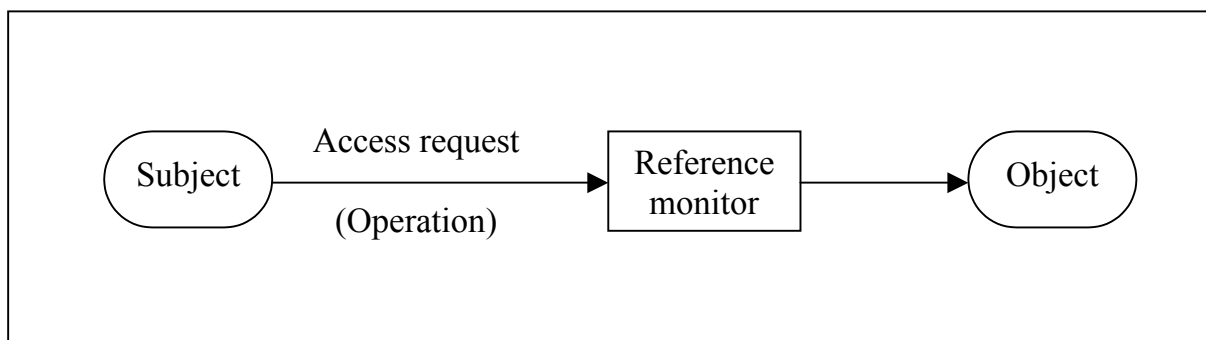


Figure 1: *The fundamental model of access control.*

Typical subjects are users and processes. In a wide approach, the possible resources on the server can be several different items. The most common items are files on a file server. Nevertheless, in our approach objects also might consist of printers, directories, and operations specific to a certain application. This approach needs a scalable solution because the number of different resources is huge. For example, as the operations might be application specific, a local administrator is needed who can set up new

permissions when new applications are installed. At the same time, the system has to be flexible enough to be manageable in a uniform manner by all administrators. If not, the security might be suffering.

Yet, not every entity in the system has either to be a subject or an object all the time. Depending on circumstances, an entity can be a subject in one access request and an object in another. The terms ‘subject’ and ‘object’ only distinguish between the active and passive party in an access request. This gives us two options for focusing control. Focus can be on either:

- what a subject is allowed to do, or
- what may be done with an object.

Traditionally, the main task of an operating system was to manage files and resources, i.e. objects. In such a setting, most access control mechanisms take the second approach. However, application-oriented IT systems, like database management systems, offer services directly to the end user. Such a system may take the first approach and incorporate mechanisms for controlling the actions of subjects.

2.1.1 Protection Rings

Our first example of access control mechanisms is the *protection rings*. They constitute a suitable way to achieve integrity in a computer system. Each subject (process) and each object is assigned a number, depending on its ‘importance’. In a typical example, these numbers could be 0,1,2,3 and processes receive their number according to the rules in Table 1.

Table 1: *Protection levels in a protection ring.*

0	operating system kernel
1	operating system
2	utilities
3	user processes

To make an access control decision, compare the subject’s and object’s number. The outcome of the decision depends on the security policy you try to enforce using protection rings. These numbers correspond to concentric protection rings, with ring 0 in the center giving the highest degree of protection, see Figure 2. In our example, subjects are only allowed to access outer objects or objects located in the same ring as the subject itself. If a process is assigned the number i , then we say the process ‘runs in ring i ’.

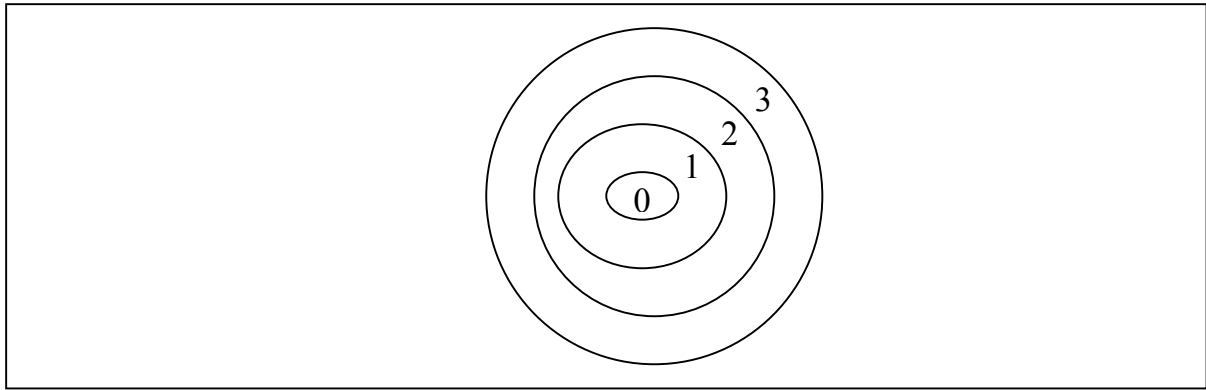


Figure 2: *Protection rings.*

2.2 Access Operations

Depending on how you look at a computer system, access operations vary from basic memory access to method calls in an object-oriented system. As we shall see later in section 2.6, comparable systems may use different access operations and, even worse, attach different meanings to operations which otherwise appear to be the same.

On the most elementary level, a subject may *observe* an object or *alter* an object. We therefore define the two *access modes*

- **observe:** look at the content of an object
- **alter:** change the content of an object

Even if observe and alter could be used to express most access policies, such policy descriptions would be too coarse-grained, making it difficult to check whether the correct policy has been implemented. Hence, you usually find a richer set of access operations and some examples of that will be given now.

In Unix the access control policies are expressed in terms of three operations:

- **read:** reading from a file
- **write:** writing to a file
- **execute:** executing a (program) file

But when applied to a directory, the access operations take the following meanings:

- **read:** list directory contents
- **write:** create, delete or rename a file in the directory
- **execute:** search the directory

As you can see, Unix controls which users can create and delete files by controlling write access to the file's directory. Other operating systems include a special **delete** operation for this purpose. Another important issue in Unix is that modifying the file's

entry in its directory changes the access rights specified for a file. In other operating systems this is done with special operations.

Windows NT is an example of an operating system that both includes a delete operation and an operation for changing permissions. The permissions used by the file system in Windows NT are: **read, write, execute, delete, change permission, and change ownership**. In NT the delete permission is stored with every file and not collectively in the directory the file belongs to.

Operations for modifying access rights are another ingredient you may want to use when setting security policies. Operations of this nature are of interest in *delegation policies*, where one subject invokes another subject and the rights of the invoked subject have to be established.

2.3 Ownership

Another important aspect is who is in charge of setting the security policy. There are two fundamental options:

- The owner of a resource declares who is allowed to have access. Such a policy may be called *discretionary* because access control is at the discretion (freedom of choice) of the owner.
- A system-wide policy declares who is allowed to have access. For obvious reasons, such a policy may be called *mandatory*.

To exemplify the first option; a user creates a file and changes the access permissions as he or she prefers. If the second option were used, the user would not be able to change the permissions because they are predetermined by the system-wide policy.

Most operating systems support the concept of ownership of a resource and consider ownership when making access control decisions. They may include operations that redefine the ownership of a resource. By that, users can share the ownership of a resource with others.

2.4 Access Control Structures

Now it is time to think of how to represent the access permissions. It is possible to represent each combination of subjects and objects. Yet, it is cumbersome if the number of objects and subjects is large. Hence, intermediate levels of control, like protection rings we mentioned earlier, are preferred because they are easier to manage. In the following sections, we will refer to:

- a set S of subjects,
- a set O of objects,
- a set A of access operations.

2.4.1 Access Control Matrix

Access rights are defined quite simply in the form of an access control matrix:

$$M = (M_{so})_{s \in S, o \in O} \text{ with } M_{so} \subset A$$

The entry M_{so} specifies the set of access operations subject s may perform on object o . Table 2 gives a simple example of an access control matrix for two users and three files.

Table 2: *An access control matrix.*

	bill.doc	edit.exe	fun.com
Alice	-	{execute}	{execute, read}
Bill	{read, write}	{execute}	{execute, read, write}

- **bill.doc** may be read and written to by Bill while Alice has no access at all.
- **edit.exe** can be executed both by Alice and Bill but otherwise they have no access.
- **fun.com** can be executed and read by both users; only Bill can write to the file.

This approach goes back to the early days of computer security. Access control matrices are also referred to as *access permission matrices*. The access control matrix is an abstract concept and not very suitable for direct implementation if the number of subjects and objects is large or if the sets of subjects and objects change frequently.

When considering implementation, there is a choice between two obvious options. Access rights can be kept with the subjects, called *capabilities*, or with the objects, called *access control lists*.

2.4.2 Capabilities

With capabilities, access rights are stored with the subjects. Every subject is given a capability, a token, which is impossible to forge and specifies this subject's access rights. This capability corresponds to the subject's row in the access control matrix. The access rights of our previous example given as capabilities are:

Table 3: *Example of capabilities, specifying subjects' access rights.*

Alice's capabilities	edit.exe: execute; fun.com: execute, read
Bill's capabilities	bill.doc: read, write; edit.exe: execute; fun.com: execute, read, write

Capabilities are strongly connected to discretionary access control, as the permissions are stored with the owner of the resource. When a subject creates a new object, it can give other subjects access to this object by granting them the appropriate capabilities.

Also, when a subject (process) calls another subject, it can pass on its capability, or parts thereof, to the invoked subject.

Capabilities have not been widely adopted even if it is an old concept. The explanation is that operating systems focus on managing objects while capabilities demand a complex security management. The reasons can also be put like this:

- It is difficult to get an overview of who has permission to access a given object.
- It is very difficult to revoke a capability because either the operating system has to be given the task or users have to keep track of all the capabilities they have passed on. This problem is particularly awkward when the rights in the capability include the transfer of the capability to third parties.

Nevertheless, capability-based access control is an interesting solution in modern distributed systems where users move physically (or virtually) between nodes in a computer network. Under such circumstances it is possible to save a storage space and communication if the access rights are stored with the clients (users) instead of each node the user connects to.

When you decide to employ capabilities, you also have to spend some thought on their *protection*. Are the capabilities stored in a safe location? If capabilities are only used within a single computer system, then it is feasible to rely only on integrity protection by the operating system. But if capabilities travel over an unsafe network, cryptographic protection is needed. Cryptographic protection of access control is examined closer in chapter 4.

2.4.3 Access Control Lists

On the contrary to capabilities, an access control list (ACL) stores the access rights to an object with the object itself. An ACL therefore corresponds to a column of the access control matrix and states which subjects may access a given object. The access rights of our previous example, given in the form of ACLs, are showed in Table 4:

Table 4: *Access control lists (ACLs).*

ACL for bill.doc	Bill: read, write
ACL for edit.exe	Alice: execute, Bill: execute
ACL for fun.com	Alice: execute, read; Bill: execute, read, write

Management of access rights based only on individual subjects can be rather cumbersome. It is therefore common to place users in *groups* and to derive access rights from the groups. In Unix, you find simple ACLs attached to files, which allow specifying basic access modes for three categories of subjects: *user*, *group*, and *others*. The category *others* matches to everyone not specified in the first two categories.

ACLs are a fitting concept for operating systems that are geared towards managing access to objects. One drawback of ACLs is the difficulty of getting an overview of a

subject's access rights. To achieve that overview all objects' ACLs has to be examined to see if the current subject is present.

2.5 Intermediate Controls

To improve management of access control we now examine alternatives to the access control matrix. It is difficult to manage a security policy expressed by such a matrix in large systems, no matter how you implement it. In particular, it is tedious and error-prone to establish that all entries in such a matrix are as desired. Moreover, access control based only on subjects and objects support a rather limited range of security policies. Further conditions may be included in the access policy. And the reference monitor needs a way to represent those extra conditions. In section 4.1, a number of those conditions are presented.

2.5.1 Groups and Negative Permissions

Groups have already been mentioned as a mean of simplifying the definition of access control policies. Users with similar access rights are collected in groups and groups are given permission to access objects. Some security policies demand that a user can be the member of one group only, others allow membership in more than one group. Figure 3 shows an ideal world where all access permissions could be mediated

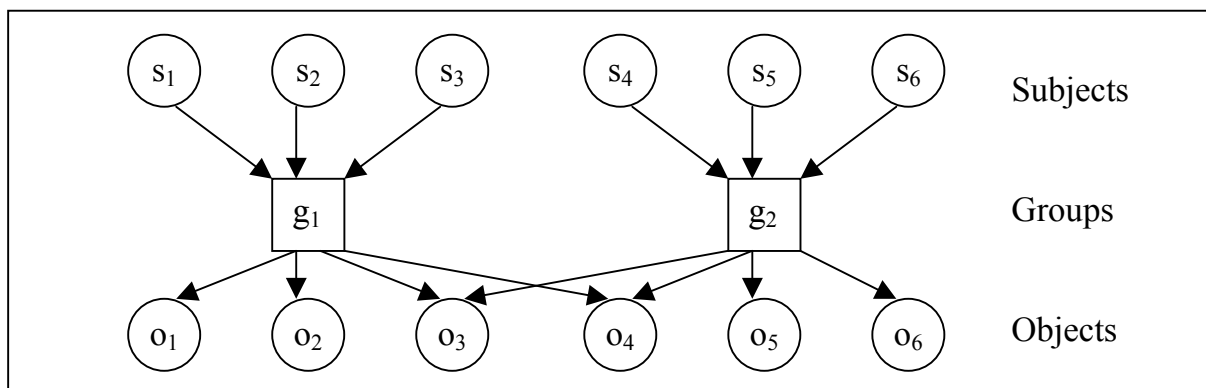


Figure 3: *Groups as an intermediate layer of access control.*

through group membership. Often, security policies have special cases where it proves convenient to give some subject permission for an object directly, or to deny a subject a permission it normally would derive from its membership in some group. A *negative permission* is an entry in an access control structure that specifies the access operations a subject is not allowed to perform. In Figure 4, subject s_1 is denied access to object o_1 and subject s_3 is granted access to object o_5 .

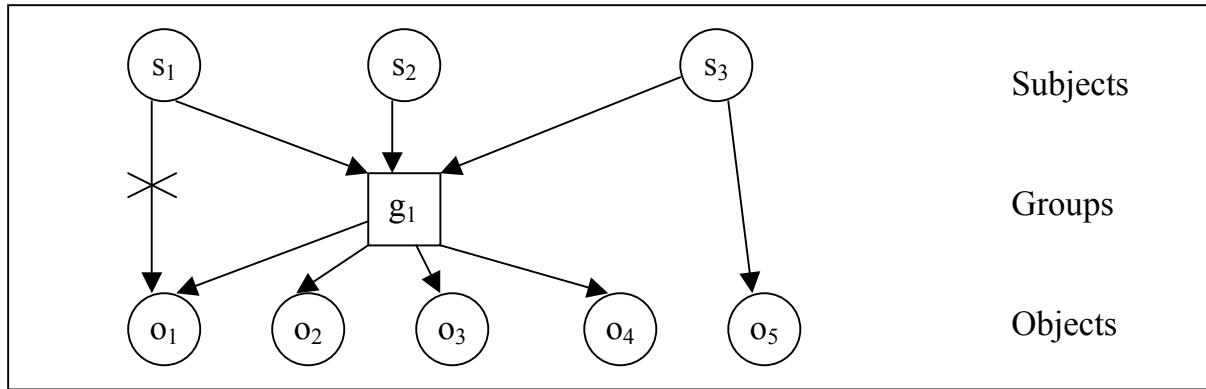


Figure 4: Access control with negative permissions.

2.5.2 Abilities

A somewhat refined mechanism is the capabilities in the VSTa microkernel [Valencia]. They are related to capabilities so let us call them *abilities*. The abilities have more internal structure and also work for objects. They form a hierarchy in which upper levels dominate lower levels. As a result, abilities can also be viewed as an extended form of protection rings.

Ability is a data structure that starts with a dot followed by a sequence of n integers, i.e. an ability is a string $.i_1.i_2.\dots.i_n$ where i_1, i_2, \dots, i_n are integers. There is no limit on the length n of such a sequence. Indeed, n may be equal to 0. Examples for abilities are $.1.2.3$, $.4$ or $.10.0.0.5$. Because of their internal structure, there exists a partial ordering on the set of abilities.

Definition: A partial ordering \leq on a set L is a relation on $L \times L$ which is

<i>reflexive</i>	for all $a \in L$, $a \leq a$ holds
<i>transitive</i>	for all $a, b, c \in L$, if $a \leq b$ and $b \leq c$, then $a \leq c$
<i>antisymmetric</i>	for all $a, b \in L$, if $a \leq b$ and $b \leq a$, then $a = b$

If two elements $a, b \in L$ are not comparable, we write $a \not\leq b$.

Abilities can be ordered through the prefix relation.

Ability a_2 is a prefix of ability a_1 if there exists another ability a_3 so that we can write $a_1 = a_2 a_3$. In this case, we write $a_2 \leq a_1$.

With this prefix relation, you can compare abilities. You would get $.1 \leq .1.2 \leq .1.2.3$ but $.1 \not\leq .4$. An access control policy could label both subjects and objects with abilities and give access if the subject's ability is a prefix of the object's ability. In this case, the ability of a superuser¹ who has access to all objects is the empty string ϵ . Thus, by not assigning an ability to a subject you would grant that subject access to all objects.

¹ Described further in section 2.6.1

Access control algorithms compare attributes of subjects and objects. It is important to check what happens if one of those attributes is missing. Fail-safe behavior would suggest that access should be denied.

2.5.3 Privileges

Privileges are very similar to grouping of subjects. However, they are worth mentioning because privilege is a common term in the literature and have a slightly different focus than user groups. Operations are central in privileges and the right to execute certain operations are collected in privileges. Even if it is the operations that are grouped, compared to subjects in user groups, the result is analogous as you can see from Figure 5.

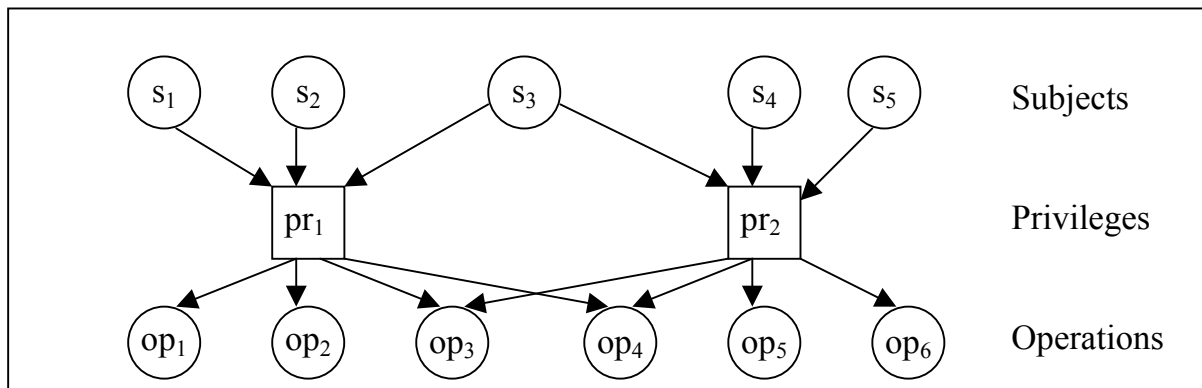


Figure 5: *Privileges between subjects and operations.*

Typically, privileges are associated with operating system functions and relate to activities like system administration, backup, mail access, or network access. To obtain two intermediate layers, user groups can be used together with privileges. The first layer would be subjects collected in groups and the second would be operations collected in privileges.

2.5.4 Role-based Access Control

Role-Based Access Control (RBAC) is a sophisticated technology of grouping subjects, in which subjects traditionally have been human users. The technology has evolved during a long period and is supported by NIST (National Institute of Standards and Technology). The concept of roles has been used in software applications for at least 25 years, but it is only within the past decade access control has emerged as a full-fledged mechanism as traditional mandatory and discretionary access control. The roots of RBAC include the use of roles in UNIX and other operating systems and privilege groupings in database management systems. The modern concept of RBAC embodies a single access control model in terms of roles and role hierarchies, role activation, and constraints on user/role membership and role set activation.

According to Ravi Sandhu [Sandhu 1997] the basic concept of RBAC is that permissions are associated to roles, and users are made members of appropriate roles, thereby acquiring the role permissions. A user can be a member of several roles and a

role can have several members. The roles can be organized in a role hierarchy where permissions are assigned and can be inherited.

The figure [Sandhu 1997] below, shows the model of RBAC. When a user logs in to a system, a session is started. A session is a dynamic connection between a user and the roles the user has access to. To regulate the organization of the access control structure there is a set of constraints; they restrict what is accepted to do in the structure. For example, a constraint may limit the number of members in a role.

A group at George Mason University [Ferraiolo et al 2000] has proposed a standard for RBAC. We will examine the components that are included in the proposed standard and how these components are related to each other.

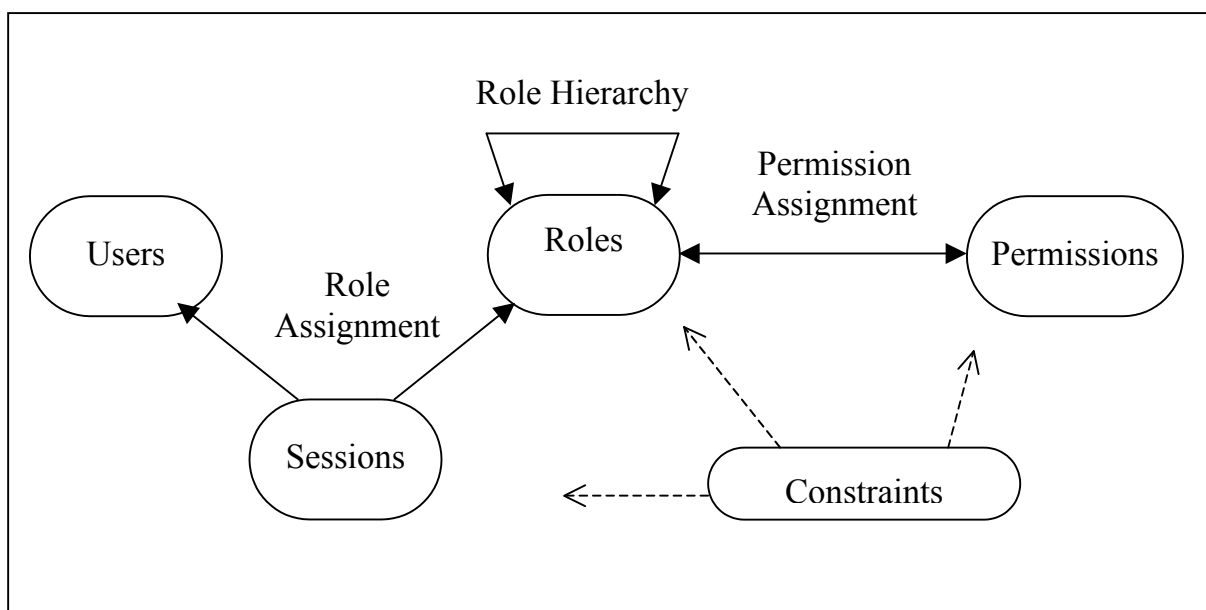


Figure 6: Model of Role-Based Access Control (RBAC).

Roles

The role concept is defined in the following way by [Sandhu 1997]:

A semantic construct around which the access control policy is formulated.

One important thing that this description points out is that organizational security policies are important to look into when the roles are defined. A security policy is a document produced by the organization that owns the system. The policy contains a high level description of how the security should be governed and maintained².

Users are granted membership into roles based on their competences and responsibilities in the organization. The operations that a user is permitted to perform are based on the user's role. One way to construct roles can be to look at the user functions in

² For further information, look up *security policy* in the glossary (Appendix B).

the organization, for example which department the user works at. But this is not necessarily the best way, sometimes it can be better to look at what task or responsibility the user has and define the roles from this.

A major difference between most implementations of groups and the concept of roles is that groups are typically treated as a collection of users and not as a collection of permissions. A role is both a collection of users on one side and a collection of permissions on the other. The role serves as an intermediary to bring these two collections together.

When there are many roles in an organization there is a probability that some of them have overlapping permissions, that is, users belonging to different roles need to perform common operations. To make role administration more efficient, RBAC includes *role hierarchies*. In the hierarchy the roles inherit permissions from each other. For example, roles can be organized to reflect authority, responsibility, and competence in an organization.

Resources

Resource in RBAC is a synonym for object, which was explained in section 2.1.

Hierarchies of resources are normally not a part of the RBAC model. But [Johansen et al 2000] believe that with visualization of the resources in a hierarchy, the task of assigning the resources to roles become easier, which hopefully leads to fewer mistakes.

There are different approaches to use when constructing a resource hierarchy. The resources could for instance be organized after their physical location or after their area of use. But if the hierarchy should consist purely of resources that have permissions, i.e. a hierarchy where each node consists of another resource, the most natural relation is the “consists of”-relation. An example is an operating system consisting of programs, which in turn consists of files.

Permissions

[Sandhu 1997] labels *permission* as an approval of a particular mode of access to one or more objects (resources) in the system. The terms authorization, access right and privilege are also used to denote permission. Permissions are always positive and award the ability to the holder of the permission to perform some action(s) in the system. The RBAC model permits a variety of interpretations for permissions, from coarse-grained, e.g. where access is permitted to an entire sub network, to fine-grained, where the unit of access is a particular operation of a particular program. In the RBAC framework, negative permissions (which deny access), are modeled as constraints rather than negative permissions.

According to [Sandhu 1997] a general model of RBAC must treat permissions as uninterpreted symbols to some extent. Each system protects objects of the abstraction it implements. Thus an operating system protects such entities as files, directories, devices, and ports, with operations such as read, write, and execute.

Constraints

As described in [Chen et al 1995], the basic idea for applying constraints is to lay out higher-level organizational policies. An example of this is that of mutually exclusive roles, i.e. a user cannot be a member of two roles that are mutually exclusive. Once certain roles are declared to be mutually exclusive, there need not be much concern about the assignment of individual users to roles. The assignment can then be delegated and decentralized without fear of compromising the higher-level policies of the organization.

A common example of mutually exclusive roles is the purchasing manager and account payable manager. In most organizations the same individual will not be permitted to be a member of both roles, because this creates a possibility for committing fraud.

Some constraints can be applied both *statically* and *dynamically*. Statically applied constraints restrict the static structure of the RBAC system and dynamically applied constraints restrict the RBAC structure during a session. Static and dynamic properties of the constraints are discussed further below.

Dynamic separation is more complicated than static separation because it must be enforced during a program session, but on the other hand it may make the system more flexible. Let us continue the example from above with the purchasing manager and account payable manager. A more flexible solution would be to allow a user membership in both roles at the same time. The result is that no user can authorize a payment (in the role as account payable manager) that he or she initiated (in the role as purchasing manager).

Another example of a user assignment constraint is that a role can have a maximum number of members. For instance, there is only one person in the role of chairman of a department. Similarly, the number of roles to which an individual user can belong could also be limited. These constraints are called *cardinality constraints*.

The last type of constraints to be mentioned here is related to the concept of *prerequisite roles*, which is based on competency and appropriateness. A user can only be assigned to role A if the user is already a member of role B. For example, only those users who are already members of the project role can be assigned to the testing task role within that project.

RBAC Summary

In RBAC, maintaining of the access control structure is certainly easier than if the users were directly connected to the resources. This is due to that the collection of users and permissions in an enterprise is transitory but the role is more stable because an organization's activities or functions usually change less frequently. The RBAC model simplifies the work of managing users' access rights in a system. When an employee is hired, moves to a new position or quits the job, the administrator's job is simply to reassign the role membership of that user. RBAC also gives direct insight into specific users permissions in the system. This helps to detect incorrectly assigned permissions.

Although structured access control of this kind is highly desirable for many applications, it is not yet supported by many operating systems. Notable exceptions, identified by [Gollmann, 1999], are the *user profiles* in IBM's AS/400 and the *global groups* and *local groups* in Windows NT. RBAC is more common in database management systems.

2.6 Operating Systems

Private networks are more homogenous compared to public networks. They are homogenous in the sense that they consist of one product-family from few manufacturers. Thus, the need for standardization of access control in private networks is not as great as in public networks.

A general pattern of security functionality can be determined in most operating systems. The following concepts are part of this pattern:

- **Accounts** – contain information, e.g. privileges about users (subjects).
- **Identification and authentication** – functionality used to verify a user's identity, allowing the system to match the user's privileges with any process started by the user.
- **Permissions** – are set on resources (objects) by the system manager or the owner of a resource. When deciding whether to grant or deny an access request, the operating system may refer to the user's identity, the user's privileges, and the permissions of the object.
- **Audit log (audit trail)** – is a log of a user's action kept in order to be able to make investigations of breaches happened earlier.
- **Installation and configuration** – it is important to start the operating system in a secure state. Furthermore, an administrator has to be able to modify security controls and therefore it is vital with configuration.
- **Default settings** – can be a major security weakness if they are inadequate.

In the following two subsections we look at how two operating systems implement access control in a private network. The purpose is to point out similarities and differences in how access control is achieved. Please note that what is stated might not be

valid for the last versions of the two operating systems. E.g., in the Unix-version Solaris 8 there is support for RBAC, which is not dealt with in the next subsection.

2.6.1 Unix

In this section we will look at access control features common in most Unix versions.

POSIX 1003 is a series of standards, produced by IEEE, regarding UNIX interfaces. The sub series *POSIX 1003.6* deals with security mechanisms as a whole and access control as a part of that.

In Unix, users are identified by *user names* and authenticated by *passwords*. Each user stores files and documents in *accounts*, which are located in personal *home directories*. User names are used together with *userID* to represent users. The username can be represented with up to eight characters and a userID consists of a 16-bit number. The userID is Unix' counterpart of the *role* concept.

An example is the userID 0, which is a role called *super user* or *root* and has special privileges. The root account is used by the operating system for essential tasks like login, recording the audit log, or access to I/O devices. Almost all security checks are turned off for the root role, which is generally used for administration purposes.

The set of userIDs is limited. However, administrators can define unlimited numbers of user *groups*. Users in one group partly share the same access restrictions.

Discretionary access control is obtained with a granularity of *owner*, *group*, and *other* (all users). Root users are not affected by any of these constellations. Unix treats all resources in a uniform manner and makes no distinction between files and devices (e.g. printers and disc drives).

Files and directories are arranged in a tree-structured file system. Each file entry in the directory is a pointer to a data structure called *inode*. An inode contains different fields and for access control the most important field is the *mode* field. The mode field states the type of file and access rights for owner, group, and other. The access rights are represented by permission bits, which are grouped in three triples that define *read*, *write*, and *execute* access for owner, group, and world, respectively.

2.6.2 Windows NT

Windows NT is POSIX compliant as Unix and includes networking capabilities. Another similarity to Unix is the distinction between *kernel mode* and *user mode*. The core operating system services run in the kernel mode and user applications in user mode.

Data is stored in *proprietary formats* and can be used by utilities to serve as an intermediate layer of control. As mentioned above, in Unix everything is treated as a resource and has a uniform discretionary access control. However, Windows NT has an

object-oriented design, which results in various object types with the possibility of unique access control for each object type.

The following components of the operating system are parts of the *security subsystem*:

- **Security Reference Monitor (SRM):** in charge of access control. The SRM is an executive component, running in kernel mode.
- **Local Security Authority (LSA):** a user mode component involved at login when it checks the user account and creates a *system access token*; the LSA is also responsible for auditing functions.

Now we turn to access control features. Our first feature is *domains*, which are used to give a group of workstations the same access configuration. Without domains, the administrator would have to configure the access rules on each and every workstation. Furthermore, the domain facilitates single sign-on so that users are only prompted once for their passwords even if they access different resources (with different access rights) during one session.

Local and *global accounts* are related to the domain concept. The local account is maintained by the workstation in an accounts database. Global accounts on the other hand, are maintained centrally in the domain database. One user can have a local and global account at the same time but will have two different security identifiers containing separate access permissions. Similarly, resources can be managed globally or locally. Typically, a resource like a printer attached to a workstation would be managed locally.

A *group* is defined as a collection of user accounts. As in Unix, members in a group partly share the same access restrictions. In Windows NT exist *local* and *global groups*³, which provide two layers of control between subjects and objects.

- The upper layer of global groups. Defined for the domain and contains only user accounts.
- The lower layer of local groups. Defined for a workstation and contains both user accounts and global groups.

Built-in accounts and *built-in groups* are similar to the userID concept in Unix because they have predefined user rights and permissions. There exist a few global built-in groups like Domain Administrators, Domain Users, and Domain Guest. Most built-in groups are however local groups like *Administrators*, *Backup Operators*, *Users*, or *Guests*. System managers are advised to stick to the built-in groups when implementing their security policies and define groups with different permission patterns only if there are strong reasons for doing so. The *Guest* account does not require a password and can be used to give users access to resources that do not require authentication. However, permissions can be given to this account like to any other user account.

³ Not to be confused with local and global accounts.

The user profile defines the user's desktop environment, in particular the programs a user is able to invoke. The user cannot change *mandatory profiles*. Mandatory profiles are a security mechanism as they can limit the utilities offered in the user's desktop environment. The administrator can set restrictions in the user profile, which define the features available in the user's desktop environment.

Unlike in Unix, administrators in Windows NT do not automatically have super user privileges that allow access to all files. Even in this finer granularity of management privileges, the Administration account is still in position to find a way around access restrictions imposed on it.

Every user, group, and machine account has a unique *security identification number* (SID), which is used for discretionary access control. The SID is constructed when an account is created and is fixed for the lifetime of the account. As pseudo-random inputs (clock values) are used in its construction, you cannot expect to get the same SID if you delete an account and then recreate it with exactly the same parameters as before. Hence, the new account will not retain the access permission given to the old account. When a domain is created, a unique SID is constructed for the domain. When a workstation or a server joins a domain, it receives a SID that includes the domain's SID. Machines use their SIDs to check whether they are in the same domain or not.

The Windows NT design follows the object-oriented paradigm. Processes, user accounts, resources, files, directories, etc., are all objects of a certain type. Discretionary access control on an object is predicated on the type of the object. For example, access control to a file differs from access control to a print queue. Each object has a *security descriptor*, giving:

- The *security ID* of the owner of the object
- A *group security ID*, used only by the POSIX subsystem
- An *access control list* (ACL)

When a subject requests access to an object, the Security Reference Monitor checks the object's ACL to determine whether the requested access should be granted or not. If no ACL exists, no checks are performed and access is granted. If an ACL exists, then for each entry the subject's SID is compared with the entry's SID.

The ACL contains entries for access control and auditing permissions. An *ACL entry* for a subject or group can be either: AccessDenied, AccessAllowed, or SystemAudit. The negative permission AccessDenied makes a difference from Unix, where all permissions are positive. AccessDenied entries are always listed first in an ACL. Each AccessAllowed entry is a list of access permissions. Access permissions are specific to the type of the object.

Access is denied if the search reaches the end of the ACL. Thus, access will always be denied if there is an empty ACL and access will always be granted if there exists no ACL.

2.6.3 Notes

This chapter ends with two notes from [Gollmann 1999] worth keeping in mind:

- “Often, operating systems store information in different places. It is important to know in which order checks are performed. Sometimes, only the first matching (access control) entry is consulted. Other times, more specific entries coming later can overrule a previous entry. Finally you have to know how the operating system reacts if it finds no entry matching an access request.”
- “Ideally, the security policy of an organization divides users with equivalent requirements into a manageable number of groups. In practice, there will always be exceptions. Therefore, mechanisms for defining exceptions, either by withdrawing or by adding permissions are useful tools, if applied with moderation.”

2.7 Summary

The basic components of access control are subject, objects, operations, and a reference monitor that either grants or denies access. The access control matrix is an access control structure, which can be implemented either as capabilities or access control lists.

Different means of intermediate access control are used to improve management. A simple way is to collect subjects into groups. The most sophisticated intermediate control we examined was RBAC, which focus on collections of permissions. These collections are used to form roles that in a later step are connected to subjects.

Both UNIX and Windows NT conform to the POSIX standard. In POSIX, access control is achieved through ACLs. In addition, both operating systems contain simple roles that give users predefined access rights and permissions. But the two operating systems differ in numerous ways. The most important ones are:

- Permissions in UNIX are only positive whereas Windows NT embraces negative permissions as well.
- All objects are treated equally in UNIX, while there exist various types of objects in Windows NT that are treated independently.

3 Security Concepts in Public Networks

This chapter serves as a preparation before discussing access control in public networks in the next chapter.

When discussing security in distributed systems, two categories of security can be identified. First we have *computer security*, which deals with the protection of resources within a computer system. The other, *network security* deals with protection of information during transmission from one system to another. The former deals a lot with access restrictions, while the latter deals mainly with cryptographic protocols.

Figure 7 illustrates network security. Two entities, A and B, communicate over an unsecured channel. The antagonist is an *intruder* who has full control over this channel, being able to read their messages, delete messages, and insert messages. The two entities trust each other. They want protection from the intruder. Cryptography allows them to construct a secure logical channel over an insecure physical connection.

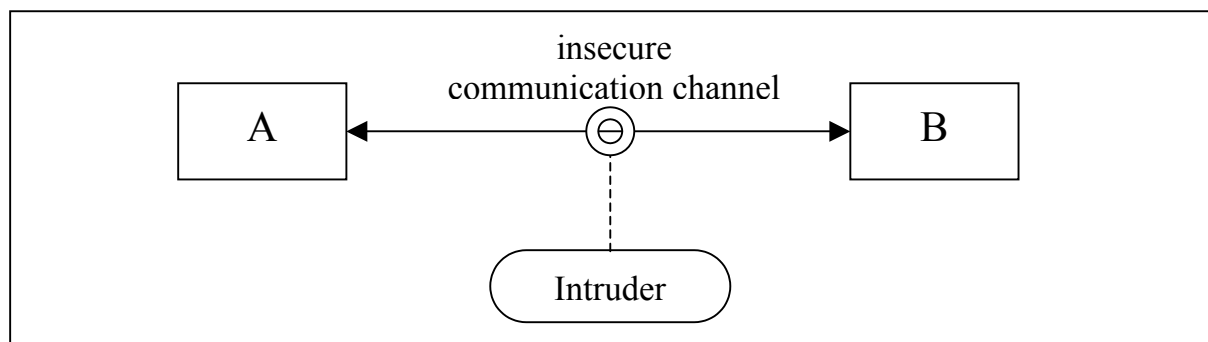


Figure 7: Public network security.

Now our focus turns from computer security, the area where access control was first established, to network security.

Access control is defined in the following way by IETF [RFC 2828]:

Protection of system resources against unauthorized access; a process by which use of system resources is regulated according to a security policy and is permitted by only authorized entities.

The following important topics of computer security are strongly related to access control:

- **Authentication** – The process of verifying the identity of an individual through the use of a username and password, digital certificate, or other means. Authentication always has to precede access control. In most solutions, the requesting subject is authenticated just before access control is performed on the object side. However, in other solutions a subject first authenticates to a third party in

order to obtain access privileges. Later, the subject only presents the privileges for the object. In this way objects do not need to be familiar with the identity of every requesting subject, only with the subject's privileges.

- **Integrity** – The property that data has not been changed, destroyed, or lost in an unauthorized or accidental manner. Continuing the previous example; it is important that the access privileges, sent from the third party (via the subject) to the object, are not tampered.
- **Availability** – The property of being accessible and useable upon demand by an authorized entity. If a requested service is not available, access is consequently not permitted.
- **Confidentiality** – The property that information is not made available or disclosed to unauthorized individuals, entities, or processes.
- **Accountability** – The property of a system that ensures that the actions of system entities and users may be traced uniquely to that entity, which can be held responsible for its actions. Accountability is needed to store information about who performed a specific action on a specific object. The information can be used later when access control is investigated.

As shown above, access control involves several issues. One mechanism that can be used to meet those issues is *cryptography*. In the following subsections, we will look at the basics of cryptography and see how cryptography can be used to achieve confidentiality, data integrity and authentication. The last subsection describes how virtual private networks can be created with the help of cryptography. But first we examine a simple example of some of the issues above.

3.1 Example of Security Issues

Authentication, access control, and accounting happen in everyday life. For instance, when you go to an ATM⁴ to withdraw money, you must first insert your bankcard and enter your personal identification number (PIN). At this point you are now authenticating yourself as someone who has the authority to withdraw money. If your card is valid, and your PIN is valid, you have been successfully authenticated and can now continue the task of withdrawing money. If you have entered an incorrect PIN, or your card has been damaged (or stolen) and the criteria cannot be validated, you will not be able to continue.

Once authenticated you will be permitted to perform certain actions, such as withdraw, deposit, check balances, and so on. Based on your identity (your bank card and your PIN), you have been preauthorized to access certain functions on your account, which include withdrawing money. Finally, once you have completed the tasks in which you are authorized to perform, you are then provided with a statement describing your transactions as well as the remaining balance of your account. The bank will also record your transactions for accounting purposes.

⁴ The Swedish word for ATM is bankomat.

3.2 Cryptography

People mean different things when they talk about cryptography. Children play with toy ciphers and secret languages. However, these have little to do with real security and strong encryption. Strong encryption can be used to protect information of real value against organized criminals, multinational corporations, and major governments. Strong encryption used to be only military business; however, in the information society it has become one of the central tools for computer security.

3.2.1 Terminology

Suppose that someone wants to send a message to a receiver, and wants to be sure that no one else can read the message. If the message is sent over a public channel, there is a risk that someone else opens the letter or taps the electronic communication.

In cryptographic terminology, the message is called **plaintext** or **cleartext**. Encoding the contents of the message in such a way that it hides its contents from outsiders is called **encryption**. The encrypted message is called the **ciphertext**. The process of retrieving the plaintext from the ciphertext is called **decryption**. Encryption and decryption usually make use of a **key**, and the coding method is such that decryption can be performed only by knowing the proper key. Figure 8 illustrates the terms above.

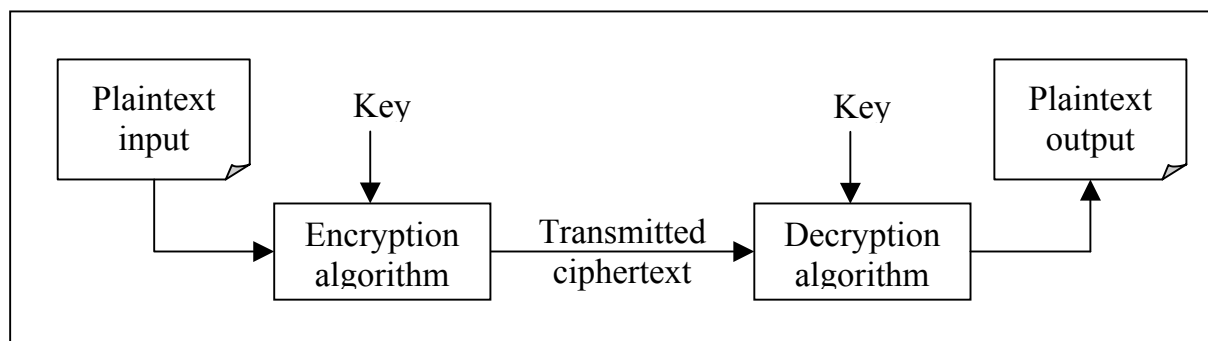


Figure 8: *Model of encryption.*

Cryptography is the art or science of keeping messages secret. **Cryptanalysis** is the art of **breaking** ciphers, i.e. retrieving the plaintext without knowing the proper key. **Cryptology** is the family name for cryptography and cryptanalysis. People who do cryptography are **cryptographers**, and practitioners of cryptanalysis are **cryptanalysts**.

3.2.2 Cryptographic Algorithms

A method of encryption and decryption is called a cipher. Some cryptographic methods rely on the secrecy of the algorithms; such algorithms are only of historical interest and are not adequate for real-world needs. All modern algorithms use a key to control encryption and decryption; a message can be decrypted only if the key matches the encryption key.

There are two classes of key-based encryption algorithms, **symmetric** (or **secret-key**) and **asymmetric** (or **public-key**) algorithms. The difference is that symmetric algo-

rithms use the same key for encryption and decryption (or the decryption key is easily derived from the encryption key), whereas asymmetric algorithms use two different keys for encryption and decryption, and the decryption key cannot be derived from the encryption key.

Asymmetric ciphers permit the encryption key to be public, allowing anyone to encrypt with the key whereas only the proper recipient (who knows the decryption key) can decrypt the message. The encryption key is also called the public key and the decryption key the private or secret key.

Generally, symmetric algorithms are much faster to execute on a computer than asymmetric ones. In practice they are often used together, so that a public-key algorithm is used to encrypt a randomly generated encryption key, and the random key is used to encrypt the actual message using a symmetric algorithm. This is sometimes called hybrid encryption.

3.2.3 Public versus Symmetric Cryptography

One drawback with symmetric cryptography is that possibly many people share the same secret. To illustrate, let us imagine a group of users that share a key to be able to obtain access to a server. Now one of these users stores his copy of the key in an insecure place and an opponent copies the key. Thus, all communication among the clients that share the key is considered insecure since the opponent can disclose messages encrypted with the key. Furthermore, it is troublesome to first stop all clients from using the stolen key. Last but not least, a new key has to be securely distributed to all clients that previously used the stolen key.

If public encryption was used, a stolen key does not have that deep impact. Consider a client, Alice, and an opponent who steals her private key. Then only Alice has to produce a new pair of keys. She keeps the private key in a secure location and publishes the public key. The next time someone wants to communicate with Alice there is no risk that the opponent will get the hands on the information because all clients that want to communicate with Alice will use the new public key that she has published. To make this system really secure the key updates have to be made on a regular basis.

Let us compare the number of keys needed if N users want to communicate in pairs. If symmetric encryption is used every pair of users needs a key. The required number of keys is $[N(N-1)]/2$. If public key encryption is used every user needs a public and a private key. Thus, the number of keys is $2N$. Thus, the number of keys that has to be generated depends significantly on who is communicating with whom and on which technique is used.

Another disadvantage with symmetric encryption is that since encryption is presumably not available prior to key distribution, network-based key distribution is not a secure option. Other options, such as a secure courier, are expensive and slow. In contrast, in public cryptography the public key can be transmitted unencrypted over insecure lines, since it is not a secret. Thus, key distribution is greatly simplified using

public key encryption. One way to handle key distribution is through a public-key infrastructure, which is discussed in section 3.4.

As mentioned earlier, symmetric encryption has the major advantage that it is computationally much faster than public-key encryption. Furthermore, symmetric encryption is handy if a message is going to be distributed to several recipients. Then the message only has to be encrypted once if all the recipients share the same key.

3.2.4 Digital Signature

In addition to use encryption with public key cryptography for obtaining confidentiality, it can also be used to create digital signatures. This is done by encrypting a message with the sender's private key. A digital signature authenticates the identity of the sender of a message or the signer of a document. Digital signatures are easily transportable and cannot be imitated by someone else. The ability to ensure that the originally signed message arrived to the receiver means that the sender cannot deny that it was sent (repudiate).

In this section we assume that the key pair, used to create and verify a signature, was created properly and that the public key is distributed without modification. Section 3.4 deals with these problems that are by no means trivial.

A digital signature can be used with any kind of message, whether it is encrypted or not, simply so that receiver can be sure of the sender's identity and that the message arrived intact.

In order to achieve data integrity and improve performance, a hash value⁵ based on the message, is signed instead of the whole message. A hash function is a function that takes an arbitrary message and transforms it into a hash value of fixed length. The value can be seen as a fingerprint of the message because it is radically smaller but still unique. By signing the fingerprint, the signing procedure takes less time and the signature takes less space.

The following example is from [Wha] and shows how digital signatures are used. Assume you are going to send the draft of a contract to your lawyer in another town. You want to give your lawyer the assurance that it is unchanged from what you send and that it is really from you.

1. Copy-and-paste the contract into an e-mail note.
2. Obtain a message hash of the contract by using special software.
3. Encrypt the hash code (not the plaintext message) using your private key.
4. The encrypted hash becomes your digital signature of the message. Note that it is unique and thus will be different for every message you send.

⁵ Further explained in the glossary (Appendix B)

At the other end, your lawyer receives the message.

1. To make sure it is intact and from you, your lawyer makes a hash of the received message.
2. Your lawyer then uses your public key to decrypt the message hash.
3. If the hashes match, the received message is valid.

This verifies the data integrity of the message, by making it impossible to change the message without detection. The different uses of the keys in public key encryption are summed up in Table 5.

Table 5: *Key usage in public key cryptography.*

To do this	Use whose	Kind of key
send an encrypted message	use the receiver's	public key
send a signature	use the sender's	private key
decrypt an encrypted message	use the receiver's	private key
verify a signature	use the sender's	public key

Now its time to examine concepts of network security on a higher level. The first one, firewalls, has not so much to do with cryptography, but is a related topic to the following concepts.

3.3 Firewalls

Firewalls are widely used to control and restrict the traffic between the private network and the public network. The word *firewall* can be seen as a generic name for a network gateway⁶ protecting the boundary of a private network. Firewalls may be implemented in either hardware or software but is typically a combination of both.

[Stallings 1999] lists the following design goals for a firewall:

- All traffic between the private network and the public network must pass through the firewall. This is achieved by physically blocking all access to the private network except via the firewall.
- Only authorized traffic, as defined in the local security policy, will be allowed to pass.

⁶ Explained in the glossary (Appendix B)

The position of a firewall is showed in Figure 9.

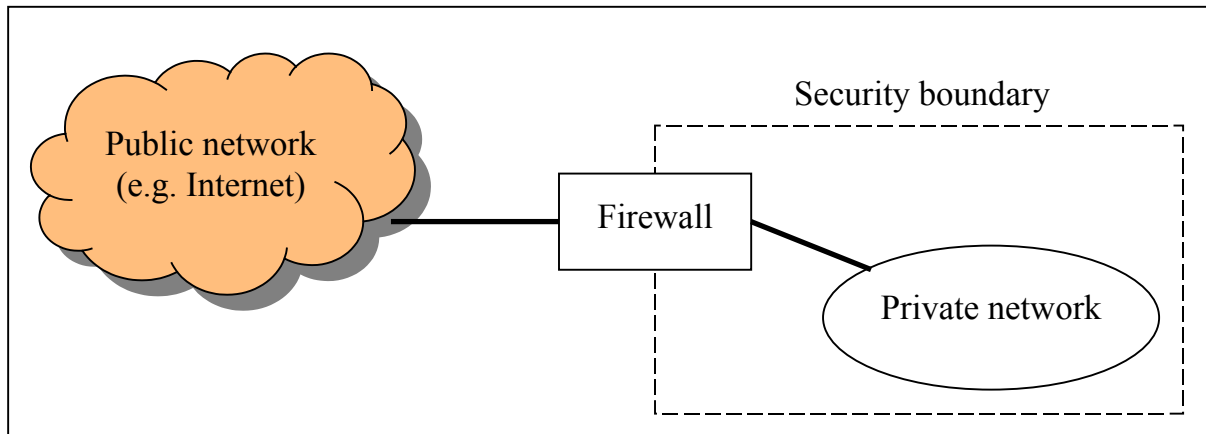


Figure 9: *Position of a firewall in a network.*

As the firewall is the only access point to the local network from outside, it can be used to hide internal information such as topology and IP-addresses of workstations.

[Stallings 1999] also points out the following techniques to enforce security by a firewall:

- **Service control** – Determines the type of services (resources) that can be accessed inbound or outbound. The firewall may filter traffic on the basis of IP address and TCP port number, provide proxy⁷ software that receives and interprets each service request before passing it on, or host the server software itself, such as a Web or mail service.
- **Direction control** – Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall.
- **User control** – Controls access to a service according to which user is attempting to access it. This feature is typically applied to users inside the firewall security boundary (local users). It may also be applied to incoming traffic from external users. However, that requires some form of secure authentication technology⁸.
- **Behavior control** – Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server.

Even if firewalls are used for access control they are rather limited. For example, attacks can bypass a firewall if the internal system has dial-in capabilities to give access to traveling employees. Modem pools are typical dial-in capabilities.

A potentially better way to achieve access control is public key infrastructure, which we will examine in the next section.

⁷ Explained in glossary (Appendix B)

⁸ Section 3.4 deals with authentication techniques.

3.4 Public-Key Infrastructure (PKI)

One way of achieving fine-grained authentication (and in a later step, access control) is to use *digital certificates*. Public-key infrastructure (PKI) is a set of security services that enable usage of public-key cryptography and certificates in a public network. The focal point is to generate and distribute keys in a secure way. But before exploring PKI, we take a closer look at digital certificates.

3.4.1 Digital Certificates

A digital certificate is an electronic “credit card” that establishes your credentials when doing business or other transactions on the Web. The certificate is commonly used for authentication and secure exchange of information in public networks. A certificate securely binds a public key to the entity that holds the corresponding private key. Certificates are digitally signed by a *Certification Authority* (CA) and can be issued for a user, a computer, or a service. The most widely accepted standard for certificates is X.509⁹, a standard defined by the ITU (International Telecommunication Union).

A digital certificate contains holder’s name, a serial number, expiration dates, a copy of the certificate holder’s public key, and a digital signature produced by the CA so that a recipient can verify that the certificate is correct.

3.4.2 Overview

PKI is defined in the following way by IETF [RFC 2828]:

A system of CAs (and optionally RAs) that performs some set of certificate management, archive management, key management functions for a community of users in an application of asymmetric cryptography.

A PKI consists of the following five main components:

- *Users* that the CA are issuing certificates for.
- *Applications* (App.) that validate digital signatures and their certification paths from a known public key of a trusted CA.
- *Certification Authorities* (CAs) that issue and revoke certificates.
- *Registration Authorities* (RAs) that guarantees the binding between public keys and certificate holders’ identities and other attributes.
- *Repositories* that store and publish certificates and certificate revocation lists (CRLs).

The relation between the components is shown in Figure 10, which is from [Aronius et al 1999].

⁹ X.509 is further described in section 4.2.2

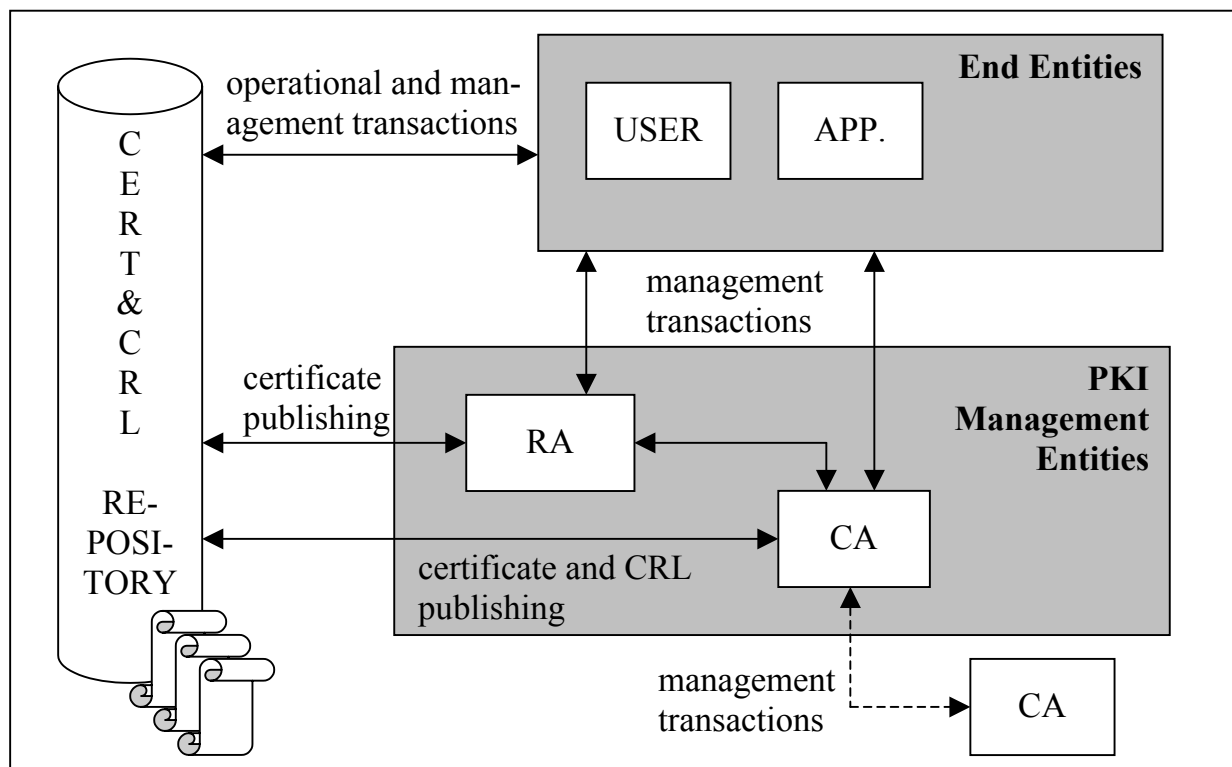


Figure 10: *Architecture of public key infrastructure (PKI).*

The core PKI functions are:

- A. To register users and issue the public-key certificates.
- B. To revoke certificates when required.
- C. To archive data needed to validate certificates at a much later time.

The quality of the keys used to create certificates is of great importance. Not all users of a PKI are capable of generating quality keys. Therefore, security policies might require the RA to generate the public and private keys. However, if a PKI client generates its own key pair, maintaining of the system integrity is simplified. The cause is that only the client possesses the private key it uses.

IETF is the main organization working on standardization of PKI. The organization has several working groups concentrating on different parts of PKI. Standard documents produced by IETF are called RFC (Request for Comments).

3.4.3 Certification Authority (CA)

A PKI is an extension of a key distribution center (KDC). The KDC simply hands out users' public keys to anyone who requests them. User A is not assured that the key is the authentic public key of user B. For instance, a malicious user attempting to steal

private information could use a man-in-the-middle¹⁰ attack to intercept the real public key and replace it with his, allowing him to decipher the data.

If users in a large community should be able to trust each other's certificates, a trusted third party is needed. The trusted third party is called CA in PKI.

The CA overcomes the weaknesses of KDC. Instead of handing out public keys, the CA issues certificates, which are distributed by a repository. The certificate is signed with the CA's own private key, making it difficult for a malicious user to successfully execute a man-in-the-middle attack.

The CA receives a certificate request from the RA, which has in turn been requested by a user (subject). The request contains the subject's public key together with additional information needed to create a certificate. The CA creates the certificate and signs it with its private key. People and certificate-using applications can confirm the certificate by verifying the signature on the certificate with the CA's public key. This only works if the entities in the community fully trust the CA.

Another function of a CA is publishing certificate revocation lists (CRLs). A CRL contains a list of invalid certificates. Various circumstances may cause a certificate to become invalid before expiration of the validation period. One reason could be a compromised key. Other circumstances could be change of name and other certificate attributes or change of relationship between the user and the CA. For example, a user changes his/her employment and no longer is authorized to access company information. A certificate must under such circumstances be revoked.

As shown in Figure 10 different CAs can be connected to form a larger PKI. CAs do this by issuing certificates to each other.

3.4.4 Repositories

After a CA has issued a certificate it is placed and stored in a *repository*. This is where users send their requests when they want to confirm other users certificates.

Repositories are used to make certificates and CRLs publicly available. Repositories are on-line entities that should have high availability requirements. However, in most cases they do not need to be trusted. A CA could operate the repository, or it could be a standalone service offered by some company.

Depending on the certificate policy, repository contents may vary, but below is a list of items that normally are published in a repository.

- **User certificates** – most user certificates should be openly available. In a big security domain including many different companies, some organizations may not wish to make the names of the employees openly available.

¹⁰ For more information, see the glossary (Appendix B).

- **CA certificates** – these should be as widely available as possible.
- **CRLs** – these should generally be openly available in the PKI.

A widely adopted standard for repositories is LDAP (Lightweight Directory Access Protocol), which is specified by IETF in [RFC 2251].

3.4.5 Registration Authority (RA)

The registration authority (RA) is a support function in the PKI, which can help the CA with different management tasks. For example, the RA handles the registration of new certificate subscribers and verifies their identities. This is a critical point in a PKI because it is important to prevent a malicious user from registering as another user.

The RA is often a physical unit to which users present ID documents to prove their identity in order to obtain a certificate. After registering and perhaps generating a key pair for the new user the RA requests the CA to create and publish a certificate for the user.

3.4.6 Summary PKI

A PKI enables users of a public network such as the Internet to securely exchange data through the use of public-key cryptography.

A number of products are offered that enable a company or a group of companies to implement a PKI. Although the components of a PKI are generally described in standards, vendors have developed a number of different approaches and services.

Concepts related to PKI are VPN, described in the next section and IPsec (IP Security) standard, described in appendix D.

3.5 Virtual Private Networks

[Kosiur 1998] defines Virtual Private Networks (VPNs) as simple as follows - *a Virtual Private Network is a network of virtual circuits for carrying private traffic*. A virtual circuit is a connection set up on a network between a sender and a receiver in which both the route for the session and bandwidth is allocated dynamically. VPNs can be established between two or more private networks, or between remote users and a private network.

A PKI can enable organizations to set up a VPN by establishing a trust relationship between their respective CAs. CAs establish the trust relationship by issuing certificates for each other. Each CA is then responsible for validating the users of its organization for participation within the VPN.

Tunnels – the “virtual” in VPN. In VPNs, “virtual” implies that the network is dynamic, with connections set up according to the organizational needs. Unlike the leased-line links used in traditional private networks, Internet VPNs do not maintain permanent links between endpoints that make up the corporate network. Instead, a connection is created between two sites when it is needed. When the connection is no

longer needed, it is torn down, making the bandwidth and other network resources available for other users. Hiding the Internet service provider and Internet infrastructure from the VPN applications is made possible by a concept called *tunneling*. Tunneling creates a special connection between two endpoints. To create a tunnel, the source end encapsulates its packets in IP packets for transit across the public network. For VPNs, the encapsulation may include encrypting the original packet and adding a new IP header to the packet. At the receiving end, the gateway removes the IP header and decrypts the packet if necessary, forwarding the original packet to its destination.

Security Services – the “private” in VPN. Equally important to a VPN, if not more, is the issue of privacy or security. In its most basic use, the “private” in VPN means that a tunnel between two users on a VPN appears as a private link, even if it is running over a shared medium. But, for business use, especially for LAN-to-LAN links, *private* has to mean more than that; it has to mean protection from prying eyes and tampering.

Although tunnels can ease the transmission of data across a public network, authenticating users and maintaining the integrity of data depends on cryptographic procedures, such as digital signatures and encryption. These procedures have to be managed and distributed with care.

3.5.1 Building Blocks

There are four main components of a VPN, the public network (e.g. Internet), security gateways, security policy servers, and certificate authorities. Not all of these components are defined or used in every current VPN product but they are common. Figure 11 shows how the components are related to each other.

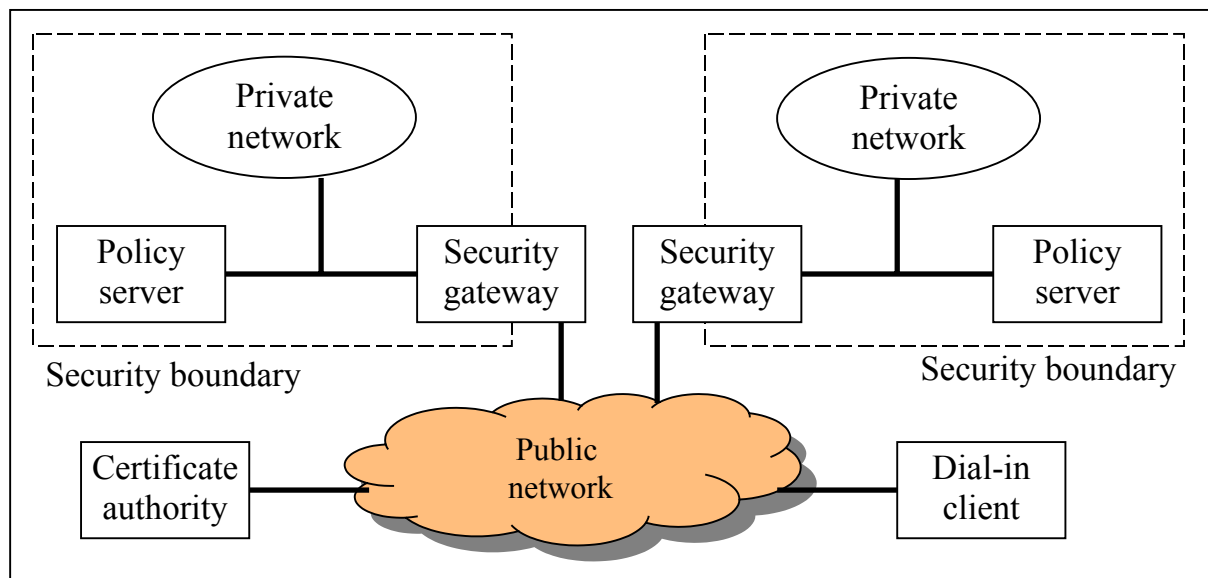


Figure 11: Components of a virtual private network (VPN).

Security gateways sit between public and private networks, preventing unauthorized intrusions into the private network. They also provide tunneling capabilities (i.e. en-

ryption and decryption of transmitted data). In general, a security gateway fits into one of the following categories: routers, firewalls, integrated VPN hardware, and VPN software. Because routers have to examine and process every packet that leaves the private network, it is quite natural to include packet encryption on routers.

Organizations can maintain their own database of digital certificates by setting up a certificate server inside the security boundary. Another solution is to use an external certificate authority that is highly specialized in managing digital certificates.

The security policy server maintains the access control lists and other user-related information that the security gateway uses to determine which traffic is authorized. Several protocols are defined for communication between gateways and policy servers.

One of these protocols is RADIUS¹¹ (Remote Authentication Dial-In User Service), which is defined for carrying dial-in users' authentication information and configuration information. A dial-in client sends authentication information to the gateway, which passes the information on the policy server. The server authenticates the client using a shared secret value and then returns to the gateway all authorization and configuration information needed by the gateway to deliver (or deny) service to the client.

The IPsec¹² standard by IETF aims at covering the whole VPN concept. IPsec defines the overall IP packet structure and security associations relative to VPN communication. However, IPsec does not include any features for fine-grained access control, as far as I know.

About 40 vendors of VPN products have unified in an organization called VPNC. The goal is to offer interoperable products. VPNC¹³ has developed a conformance test to determine if a product is interoperable or not.

3.6 Summary

There are many standards covering cryptography algorithms including symmetric and public techniques.

Public key infrastructure has also been standardized in mayor parts but is still evolving. A great number of papers published during the recent months, for example by IETF, deal with PKI on a detailed level. These papers are most probably going to lead to new standardizations.

Virtual private networks are less standardized due to the very broad concept. It is therefore difficult to point out any standard covering the whole concept. Nevertheless, IPsec is an evolving VPN standard and might in the future cover all aspects of VPNs.

¹¹ Further explained in Appendix D.

¹² Further explained in Appendix D.

¹³ Further explained in Appendix D.

4 Access Control in Public Networks

After the overview of security in public networks in the preceding chapter we return to access control. This time, our focal point is access control in public networks. As mentioned earlier new requirements are added in public networks and they are worth repeating:

- Insecure communication channel
- Heterogeneous environment
- Huge number of entities (subjects and objects)

The second point, heterogeneous environment, is the reason why we look for a standard solution. A standard offers products by different suppliers a way to collaborate and to be interoperable.

This chapter starts off with an introduction to the current SMC by Sectra. After that, conditions introduced by the insecure communication channel are pointed out. Then we examine different techniques to achieve access control under these new conditions. The chapter ends by identifying access control mechanisms in three proprietary products.

4.1 Security Management Center (SMC)

To be able to see what is needed in Sectra's Security Management Center (SMC) this section gives a brief description of the current product.

To use our previous knowledge we can describe the SMC as a counterpart to a certification authority in PKI because the SMC also works as a trusted third party. Nevertheless, a great difference is that symmetric cryptography is used. Hence, the SMC does not issue certificates. Its main purposes are to generate and distribute keys and tickets to the clients.

As shown in Figure 12, an administrator is located inside the local network. Communication with clients and other SMCs is done through external network connections based on the IP protocol.

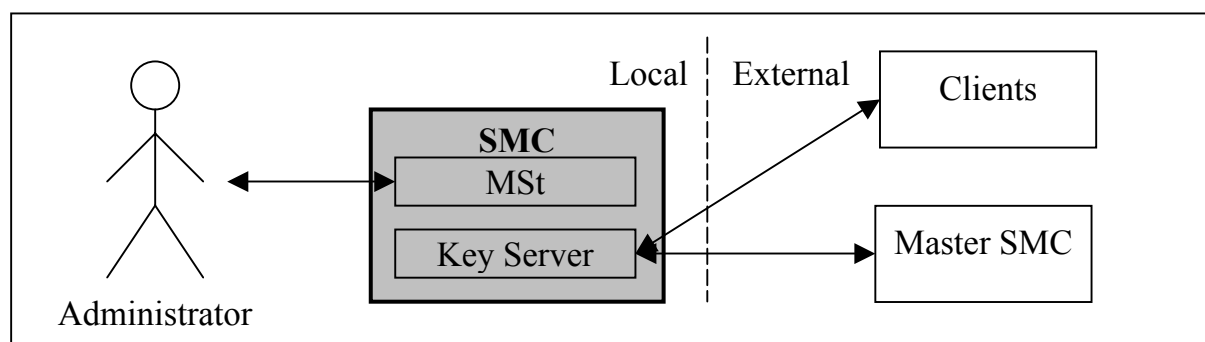


Figure 12: *Environment of the SMC.*

The SMC consists of a management station (MSt) and a key server. The management station contains user interfaces for configuration and management of the SMC. Administrators use the MSt to configure the key server and its database containing information about clients.

The key server contains all cryptographic functions together with databases of keys, clients, groups of clients and access rules. A client group consists of clients and possible other client groups. The access rules are not very detailed, they only specify who is allowed to communicate with whom. Hence, they do not specify any details of what is actually accessed on the object side.

4.2 Access Conditions

In our scope access conditions can be arbitrary. However, they have to be formulated in a structured way so an access provider can determine if they are fulfilled or not. In most organizations, access conditions are specified in a security policy document together with other security mechanisms, which provide services to protect sensitive and critical system resources.

I formed the following list of eight conditions with the inspiration of literature and discussions with my supervisors. Conditions 1) to 3) always have to be treated, independent of internal or external communication.

1) Is the client valid?

First of all the user has to be authenticated as someone who has access rights to the current resource. There are several standardized ways to implement the authentication process, e.g. as it is done in Kerberos¹⁴. The users do not have to be granted access individually. They could also be collected into groups and be given common access rights.

2) Which resource is requested?

The access permission is depending on which resource the user is asking for at the server. Different resources on the same server can have various restrictions regarding access.

3) Which operation is requested?

The access control depends on what operation the user wishes to perform on the resource. (E.g. start, stop, read, and write)

The insecure communication channels in public networks cause conditions 4) to 7). However, it is possible that the outcome of the previous conditions makes these new conditions superfluous. For example, if the requested resource is a homepage without any security restrictions there is no need to check if the user is securely authenticated

¹⁴ Described in Appendix D.2.3

or communicating through an encrypted channel. Then the following conditions might be skipped.

4) Does the client reside inside or outside the private network?

The location affects the access decision in the following way. If the user is located inside the corporate network an encrypted channel perhaps is not needed. On the contrary, if the user calls from outside it might be important to establish an encrypted channel.

5) Is the communication channel encrypted?

The communication channel through which the client tries to establish a connection with the server is not encrypted. The server denies access because it would be unsuitable to send sensitive information in a way that eavesdroppers could access it.

6) Which authentication method is used?

Whether a password or a PKI certificate authenticates the user has an impact on the access verdict because they are not equally secure.

7) How secure is the client's environment?

If for example the operating system at the client side is less secure than on the server side, the server should not let the client download any sensitive data that might be tampered with by unauthorized users on the client side. Quite obvious, the data at a secure node in a network is not secure any more if the data is transferred to another node which cannot protect it properly. The IP address can be controlled and determine access.

Another condition that might be interesting in both private and public networks is:

8) What time of day is it?

Perhaps the organization wants to restrict the access to certain time periods. For example, an officer in the military should obtain access to sensitive data only at the working hours and not when he is off duty.

4.2.1 Grouping of access conditions

Some of the access conditions are related. I will try to determine which ones are possible to group together and executed in similar ways. A pattern can be recognized if we consider the access conditions from a physical client/server¹⁵ point of view.

To start with, conditions **2** and **3** are located at the server side. Furthermore, condition **3** is depending on condition **2** because without a corresponding resource, the operation does not make sense.

¹⁵ For an explanation, see the glossary (Appendix B)

Conditions 1, 4, 6, 7, and to some extent also condition 5 are related to the client side. Thus, they can be grouped together. In addition, they are varyingly detailed, more or less in the following order: 1, 4, 5, 6, and 7.

Finally, condition 8 is not physically related to any other condition since time adds another dimension. However, the server "decides" what times access is permitted to clients. Hence, condition 8 can be grouped with conditions 2 and 3.

4.3 Standardization Work

No standard has been found that satisfies all our requirements, i.e. checks all the conditions described above. However, some products satisfy them all. These products are using related standards, which each satisfies a subset of the requirements. In this section we take a look at the related standards.

4.3.1 LDAP

Lightweight Directory Access Protocol (LDAP) is a software protocol for enabling anyone to locate organizations, individuals, and other resources such as files and devices in a network. LDAP in its current version does not support access control. However, *LDAP Extension* (an IETF working group) is developing access control for LDAP and has collected requirements in [RFC 2820].

LDAP is a "lightweight" (smaller amount of code) version of Directory Access Protocol (DAP), which is a part of X.500, a standard for directory services¹⁶ in a network. LDAP is lighter because in its initial version it did not include security features. LDAP originated at the University of Michigan and has been endorsed by at least 40 companies. Netscape includes it in its latest Communicator suite of products. Microsoft includes it as part of what they call Active Directory in a number of products, including Outlook Express. Novell's NetWare Directory Service interoperates with LDAP. Cisco also supports it in its networking products.

In a network, a directory tells you where in the network something is located. On TCP/IP networks (including the Internet), the domain name system (DNS) is the directory system used to relate the domain name to a specific network address (a unique location on the network). However, you may not know the domain name. LDAP allows you to search for individuals without knowing where they are located (although additional information will help with the search).

An LDAP directory is organized in a simple tree hierarchy consisting of the following levels:

- The root directory (the starting place or the source of the tree), which branches out to
- countries, each which branches out to
- organizations, which branch to

¹⁶ Explained in the glossary (Appendix B).

- organizational units (divisions, departments, and so forth), which branches out to (includes an entry for)
- individuals (which includes people, files, and shared resources such as printers)

An LDAP directory can be distributed among many servers. Each server has a replicated version of the total directory that is synchronized periodically. An LDAP server is called a Directory System Agent (DSA). An LDAP server that receives a request from a user takes responsibility for the request, passing it to other DSAs if necessary, but ensuring a single coordinated response for the user.

4.3.2 X.509 – Digital Certificate

X.509 is an International Telecommunication Union (ITU) recommendation and is part of the X.500 series. X.509 defines the most recognized public-key certificate.

X.500 is a directory service, which is a server or distributed set of servers that maintain a database of information about users. The information includes mapping from user name to network address, as well as other user attributes. X.500 is often substituted for LDAP because of efficiency reasons. LDAP is less complex since it has left out some parts of X.500 that are rarely used.

X.509 prescribes also a framework for authentication services, which are using the X.500 directory. The directory may serve as a repository of public-key certificates. Each certificate contains the public key of a user and is signed with the private key of a trusted certification authority.

X.509 was initially issued in 1988 and is based on the use of public-key cryptography and digital signatures. The standard does not dictate the use of a specific algorithm but recommends RSA. The digital signature scheme is assumed to require the use of a hash function. Again, the standard does not dictate a specific hash algorithm.

The heart of the X.509 scheme is the public-key certificate associated to each user. Some trusted Certification Authority is assumed to create the user certificates and place them in a directory. The directory server itself is not responsible for the creation of public keys or for the certification functions; it merely provides an easily accessible location for users to obtain certificates.

A X.509 certificate consists of the following fields:

- **Version** – differentiates among successive versions of the certificate format.
- **Serial number** – an integer value, unique within the issuing CA that is unambiguously associated with this certificate.
- **Signature algorithm identifier** – the algorithm used to sign the certificate, together with any associated parameters. Because this information is repeated in the signature field at the end of the certificate, this field has little, if any, utility.
- **Issuer name** – X.500 name of the CA that created and signed this certificate.

- **Period of validity** – consists of two dates: the first and last dates on which the certificate is valid.
- **Subject name** – the name of the user to whom this certificate refers. That is, the certificate certifies the public key of the subject who holds the corresponding private key.
- **Subject's public-key information** – the public key of the subject, plus an identifier of the algorithm for which this key is to be used, together with any associated parameters.
- **Signature** – covers all of the other fields of the certificate; it contains the hash code of the other fields, encrypted with the CA's private key. This field includes the signature algorithm identifier.

Like user identities, certificates are used for two purposes. They can identify an entity associated with a cryptographic key or they can specify the access rights for the holder of a cryptographic key (possibly without identifying the holder's identity). Experiments have been conducted with X.509 certificates that have been extended to include fine-grained access rules. Two examples are found below – Role-Based Access Control and Trust Management.

4.3.3 Role-Based Access Control

The Role-Based Access Control model, described in section 2.5.4, was not constructed with a public computer network in mind. However, to show that the model is adaptable to a public network an experiment has been performed by a group of researchers, see [Park et al 2000]. The work was partially supported by the National Security Agency in the US and one of the members in the research-group was Ravi Sandhu who is an influential person in the area of RBAC. The experiment integrated and extended well-known network technologies, such as X.509, SSL¹⁷, and LDAP, providing compatibility with current Web technologies. Access control was added to X.509 certificates by using available extension fields.

Different architectures were developed and made to work, which shows that RBAC is a possible solution to our problem.

4.3.4 Trust Management

A traditional system-security approach to the processing of a signed request for action treats the task as a combination of authentication and access control.

The receiving system first determines *who* signed the request and then queries an internal database to decide *whether* the signer should be granted access to the resources needed to perform the requested action. [Blaze et al 1999] believe that this is the wrong approach for today's dynamic, internetworked world. In a large, heterogeneous, distributed system, there is a huge set of people (and other entities) who may make requests, as well as a huge set of requests that may be made. These sets change often and cannot be known in advance. Even if the question "who signed this re-

¹⁷ Described in Appendix D.

quest?” could be answered reliably, it would not help in deciding whether or not to take the requested action if the requester is someone from whom the recipient is hearing from the first time.

The right question in a scattered, rapidly changing network becomes “is the key that signed this request *authorized* to take this action?”

In the area of Trust Management, *policy* is used for access control. IPsec¹⁸ is the standard suite of protocols for network-layer confidentiality and authentication of Internet traffic. The IPsec protocols, however, do not address the policies for how protected traffic should be handled at security endpoints. *Trust management* introduces an efficient policy management scheme for IPsec according to [Blaze et al 2001].

IPsec does not address the problem of managing the policies governing the handling of traffic entering or leaving a host running the protocol. By itself, the IPsec protocol can protect packets from external tampering and eavesdropping, but it does nothing to control which hosts are authorized for particular kinds of sessions or to exchange particular kinds of traffic. In many configurations, especially when network-layer security is used to build firewalls and VPNs, such policies may necessarily be quite complex. There is no standard interface or protocol for controlling IPsec tunnel creation, and most IPsec implementations provide only rudimentary, packet-filter-based and ACL-based policy mechanisms.

The crudeness of IPsec policy control, in turn, means that in spite of the availability of network-layer security, many applications are forced to duplicate at the application or transport layer cryptographic functions already provided at the network layer.

The notion of *trust management* is introduced in [Blaze et al 2001] as a system, which provides a standard interface that applications can use to test whether potentially dangerous actions comply with local security policies. More formally, trust-management systems are characterized by:

- A method for describing *actions*, which are operations with security consequences that are to be controlled by the system.
- A mechanism for identifying *principals*, which are entities that can be authorized to perform actions.
- A language for specifying application *policies*, which governs the actions that principals are authorized to perform.
- A language for specifying *credentials*, which allow principals to delegate authorization to other principals.

The trust-management approach to distributed-system security was developed as an answer to the inadequacy of traditional authorization mechanisms. Trust-management

¹⁸ Described in Appendix D.

engines avoid the need to resolve "identities" in an authorization decision. Instead, they express privileges and restrictions in a programming language.

According to [Blaze et al 1999] there are a number of fundamental reasons that ACLs are inadequate for distributed-system security:

- Authentication
- Delegation
- Expressibility and extensibility
- Local trust policy

[Blaze et al 1999] believes that these points constitute a forceful argument that X.509 and generally the use of identity-based public-key systems in conjunction with ACLs are inadequate solutions to distributed security problems. Even modern ACL-based systems like DCE¹⁹ fall somewhat short of satisfyingly addressing the extensibility, expressibility, and delegation issues.

The KeyNote Language

KeyNote is a trust-management system and has been developed since 1997 and has evolved to the point of supporting a wide variety of applications, ranging from IPsec policy control to electronic payment systems. The current version of the KeyNote language is described in [RFC 2704].

KeyNote provides a simple language for describing and implementing security policies trust relationships, and digitally signed credentials. KeyNote allows the creation of sophisticated security policies and credentials in which entities (which can be identified by cryptographic public keys) can be granted limited authorization to perform specific kinds of trusted actions. When a "dangerous" action is requested of a KeyNote-based application, the application submits a description of the action along with a copy of its local security policy to the KeyNote interpreter. KeyNote then "approves" or "rejects" the action according to the rules given in the application's security policy.

Thus, the access decision depends on the key that signed the privileges and not the identity of the key-owner. This is the goal of trust management – to trust the privilege-signing key. Therefore determining the identity of the subject is needless.

4.4 Products

This will be a case study of three example products²⁰ that implement access control mechanisms. They are chosen arbitrarily, the only requirements being that they offer fine-grained access control and are aimed for public networks. The products are examined from these points of view:

¹⁹ Described in section 4.3.2 and in Appendix E.

²⁰ Each of the three products is further described in Appendix E.

- **Q1:** What kind of access control is available?
- **Q2:** How is access control obtained?
- **Q3:** Which standards are used in the products?

4.4.1 DCE by the Open Group

- **A1:** Individual and group-based access control. Objects can inherit ACLs from superior objects. Time condition is available.
- **A2:** Inheritance of ACLs is achieved through a tree-hierarchy called Sparse ACLs. The time condition is achieved by time-stamped tickets that clients receive from a Ticket Server. The client sends the ticket to the requested resource.
- **A3:** Kerberos, LDAP, POSIX, SSL, X.509.

DCE takes a very broad approach and is therefore somewhat difficult to analyze. From the standards used, we see that both symmetric (in Kerberos) and public-key cryptography (in X.509) can be used. POSIX render a possibility to form access rules in a standardized manner. LDAP and X.509 are cornerstones in the authentication mechanisms.

4.4.2 SelectAccess by Baltimore Technologies

- **A1:** Offers role-based access control together with these conditions: authentication method, encrypted channel, and inside/outside private network. Access control is administrated with a graphic access control matrix. It is a scalable matrix since it is implemented as a two-dimensional tree.
- **A2:** The Policy Validator (similar to a firewall) performs the access control. The Policy Validator calls a Directory Server (similar to repository in PKI) in order to find rules corresponding to the current object.
- **A3:** LDAP, RADIUS, X.509.

As in DCE, LDAP and X.509 are used as authentication mechanisms. The access control and hierarchy of user roles is implemented in a proprietary manner.

4.4.3 VPN-1 by Check Point

- **A1:** Offers individual and group-based access control with several of the desired conditions. E.g. encrypted communication channel, authentication method, and time-of-day.
- **A2:** A security policy is defined centrally and is distributed to multiple enforcement points throughout the network. A management server maintains databases located at a firewall. The databases include security policy, accessible objects, and user definitions.
- **A3:** IPsec, LDAP, RADIUS, X.509

Once again, LDAP and X.509 are cornerstones in the authentication mechanisms. The access control functionality is solved in a proprietary way.

4.5 Summary

I have not been able to find any standard that matches all the conditions on access control listed in section 4.2. However, techniques supporting access control have been standardized. In addition, various products implement access control that fulfills most parts of our requirements.

LDAP and X.509 constitute important standards of an authentication service. X.509 can be extended to include access control as in the experiment with RBAC. Trust Management takes a step further and omits the authentication part and focus on access privileges. However, some authentication is still needed and is performed at an early phase of the procedure.

To compare the three products (SelectAccess, DCE, and VPN-1) was a difficult task since they use different terminologies. In addition, I have only read the companies own information, which naturally never contain any revolutionary details about their implementations. Hence, my knowledge base is rather limited. However, the papers revealed that the products have much in common with PKI, such as a trusted third party like a CA with the ability to issue X.509 certificates. User information is stored in LDAP-servers, which correspond to repositories in PKI. Although all the three products offer fine-grained access control they have implemented the functionality in different ways. None of them have chosen extended X.509 certificates as in the RBAC experiment. Neither have they used the Trust Management approach.

5 Prototype

Now we take a look at the prototype, which was developed during the last part of the thesis. The goal of the prototype was to examine how access rules could be defined through a graphical user interface (GUI). The produced GUI is definitely not an ultimate solution but lead to a deeper understanding of the difficulties involved in setting up access rules. An evaluation was performed to see how users experience the prototype.

5.1 Background

The motivation for developing this prototype is the importance of defining access rules in an intuitive and flexible way. If an administrator is misled into defining incorrect access permissions by an ambiguous GUI, security might be suffering. To start with, the whole range of possible access rules and conditions have to be considered. The next question is how to computerize the rules so they can be interpreted and maintained by a system on its own.

The concepts introduced in section 2.1 (subjects, objects, and operations) form a basis for the prototype. Added to those are conditions, which were described in section 4.2. As new conditions might show up, the prototype has to allow new conditions to be defined.

Role-based access control, showed in section 2.5.4, is a sophisticated way to improve subject management and is definitely desired a real product. However, due to time limits, the prototype is restricted to a simplified version of roles.

There is a need for verifying defined rules. Hence, a test tool is desirable. The test tool should take defined rules and fulfilled conditions as input and determine if access is permitted as output. A necessary simplification is that no underlying functionality is used to control if conditions are fulfilled. Thus, neither certificates nor any network implementation is developed. The focus is on central administration of access rules.

My supervisors at Sectra proposed VB (Visual Basic) to be the implementation environment. Additionally, I have decided to use a Microsoft Access database to store the different entities and their attributes.

5.2 Design

As mentioned earlier the entities subject, object, operation, and condition are central in the prototype. They are components of each rule and are depicted in Figure 13.

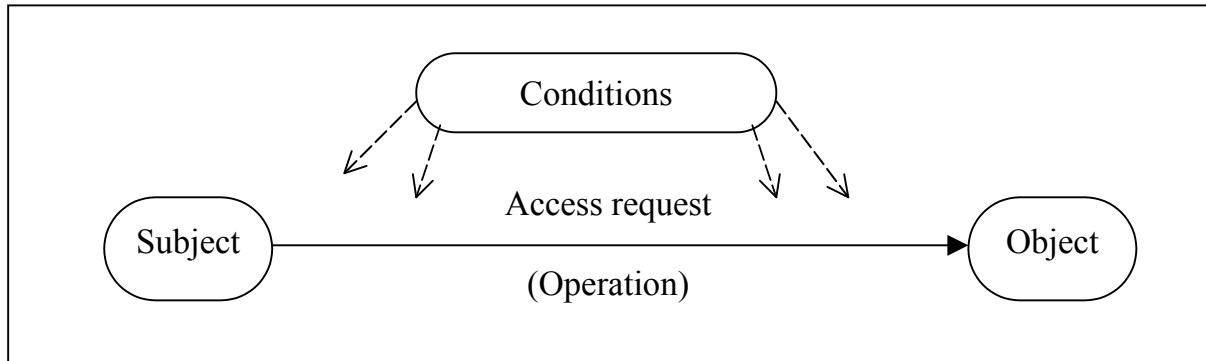


Figure 13: *Components of a rule.*

Subjects are categorized into individuals and roles. Roles are defined in a tree hierarchy where every level inherits permissions from superior roles. The roles are mainly incorporated in order to decrease the number of rules that has to be defined.

Another feature, which also reduces the number of rules, is negative rules. For example, consider a role that is granted access to a number of objects and one of the individuals in the role is not supposed to get access to one of those objects. Then a negative rule is defined, which denies access to the object, for the specified individual.

Objects are organized in a tree structure similar to subjects. However, nodes in the object tree do not inherit permissions from superior nodes in the tree. This restriction is also due to time constraints and reduces the complexity but unfortunately also the flexibility of the system. In order to identify the nodes in the subject and object trees, they are given unique ID-numbers based on their location in the tree. This is similar to abilities, described in section 2.5.2. An example of a small tree of nodes and corresponding ID-numbers is given in Figure 14.

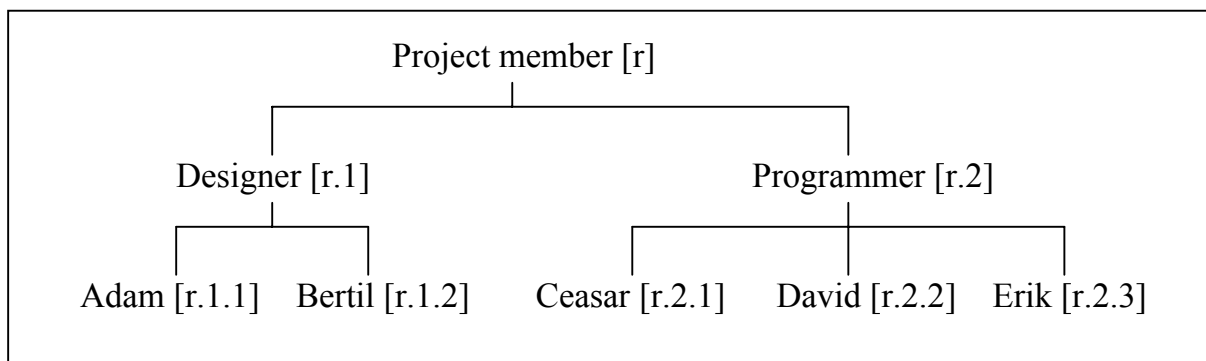


Figure 14: *A tree sample with nodes and corresponding ID-numbers.*

Operations and conditions are collected in separate lists. In section 4.2.1, conditions are grouped together from a physical point of view. However, that feature is not considered in the design of this prototype with reference to time bounds. Operations and conditions are distinguished through their unique names.

In the Access database there are tables for subjects, objects, operations, conditions, and rules. A separate table is used for determining which operations can be performed on each object. The window used for editing the separate entities and connecting operations to objects is showed in Figure 15.

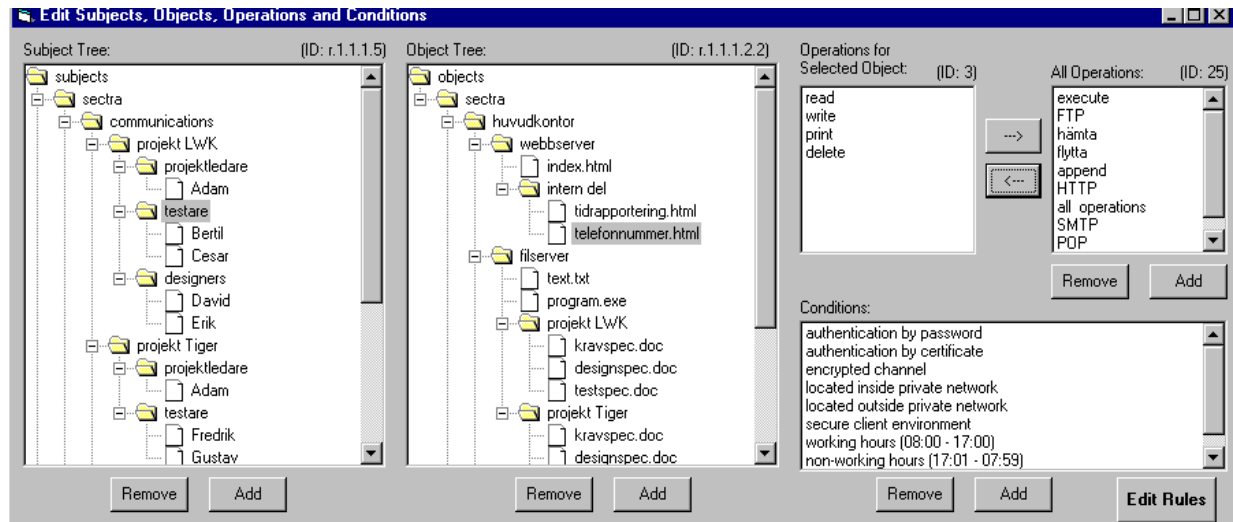


Figure 15: Managing subjects, objects, operations, and conditions.

Rules are defined in a separate window where the user selects which entities that should be included and then chooses to create a positive or a negative rule. Each rule contains only one condition. This window also contains the test tool. The test tool works similar to defining new rules with the exception that the user presses a “Run Test” button instead of adding a rule. A test is performed in the following way. The test tool starts to look for rules matching the selected subject. If no rule matched, another search is performed, but this time on the role that the subject belongs to. If still no rule is found, the searching goes on with superior roles until the top node (i.e. the root) is reached. There are two options in the test tool:

- **Inheritance** – determines if the test tool should omit rules defined for subjects superior to the selected one.
- **Multiple conditions** – this feature only affect the result if multiple conditions are selected as input to the test tool.
 - If ‘OR’ is chosen, the test tool will permit access if at least one positive rule matches one of the input conditions and no negative rule blocks the matching positive rule.
 - If ‘AND’ is chosen, the test tool will permit access if all the positive rules matching the selected subject, object, and operation are fulfilled by the set of selected conditions and no negative rule blocks any of the matching positive rules.

The inheritance switch option is only included for experimental usage. But the second option with multiple conditions is highly desirable because it makes the rules more specific. Obviously, in a real system this option should be available when the rules are defined and not only at the point of verification, as in our test tool. The window containing the test tool and positive and negative rules is depicted in Figure 16.

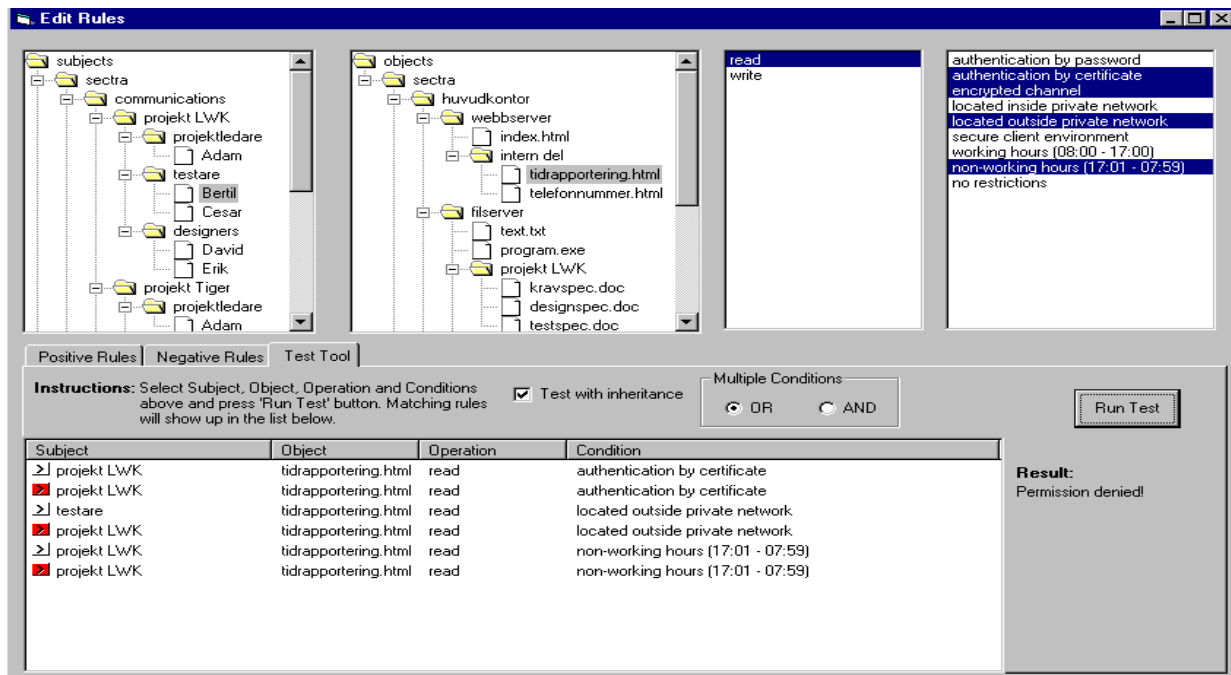


Figure 16: Managing positive and negative rules and a test tool.

5.3 Evaluation

Only employees at Sectra were involved in the evaluation of the prototype. Among them were people who developed the current SMC and a corporate administrator who deals with access control frequently. They found the prototype easy to use and a good way to experiment with different scenarios. During the evaluation several issues and ideas of improvements came up. The most important comments are summarized in the following list.

- An individual should be able to belong to several roles. In the current solution, an individual is restricted to membership of one role (and superior roles, if existing).
- The test tool should contain a feature in which it is possible to see all rules affecting a selected subject, independent of objects, operations, and conditions.
- It is desirable that a rule can contain multiple conditions, which are combined by 'AND' and 'OR' separators.
- Inheritance in the object tree is attractive although it leads to immense complexity.

The complexity introduced by inheritance in the object tree is due to the combination of positive and negative rules together with inheritance in the subject tree. Consider the following example, which is showed in Figure 17. The role *programmer* is denied access to a file called *secret.txt* through a negative rule but *David* (who belongs the role *programmer*) is granted access to the directory *public* (in which *secret.txt* is located) through a positive rule.

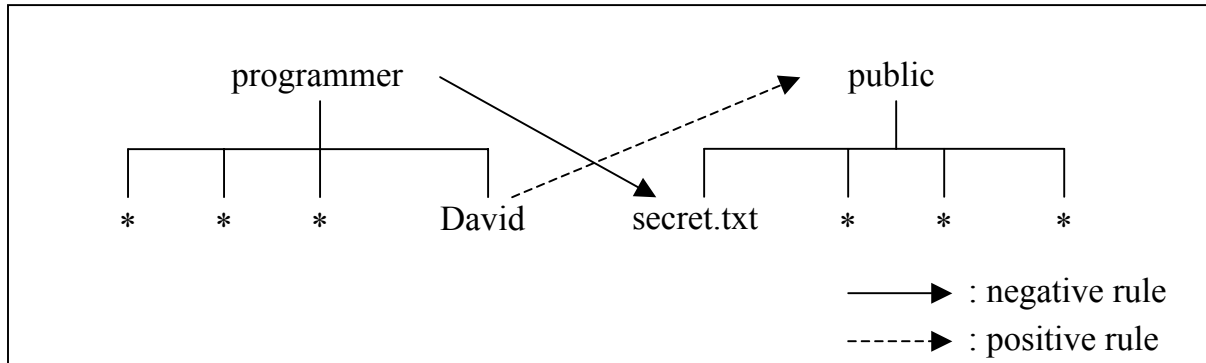


Figure 17: Crossing positive and negative rules.

This is an ambiguous situation if the rules are not given different priorities. One point of view is to focus on the security of the objects. Then access is denied to an object if there exists any negative rule including the current object and subject. On the other hand, if the possibility to give individuals extra privileges is considered valuable, then crossing positive rules are given higher priority than negative ones.

5.4 Summary

The prototype was successful in the way that it resulted in intensive discussions how access rules can be set up. Several new topics came up during the evaluation. When you have a user interface it is much easier to find new topics, which otherwise are hard to notice.

It is important to remember that this prototype only dealt with administration and assumes that other functions assure that the access rules are managed correctly.

6 Conclusion

Some general conclusions open this chapter. After that, the current version of SMC is evaluated. We finish off by a look into the future of access control.

6.1 General Conclusions

The basic components of access control are subject, objects, operations, and conditions. Role-based access control is the most sophisticated method found, for improving subject management.

The operating systems UNIX and Windows NT include various efficient access control mechanisms. Both operating systems conform to the POSIX standard, which uses ACLs as an intermediate level of access control. In addition, the operating systems contain simple counterparts to roles.

Several security issues are related to access control. Cryptography serves as a mechanism to meet those issues. Public key cryptography is a convenient technique to achieve confidentiality, authentication, and data integrity. The IETF has standardized the usage of identity certificates in a public-key infrastructure (PKI). PKI can be used to fulfill the three issues mentioned earlier. Experiments using attribute certificates have attained fine-grained access control.

The insecure communication channels of a public network introduce new access conditions. These were exemplified in section 4.2. An ultimate model is not limited to these conditions but should be flexible enough to undertake arbitrary conditions. No standard has been found that fulfils arbitrary conditions. One reason is that it is very hard to implement such flexible systems. Techniques supporting access control have been standardized and in addition, various products implement access control that fulfills most parts of our requirements.

I consider fine-granular access control in a distributed environment to be a matter of scaling. The amount of subjects and objects in a public network is much larger than in a private network. The directory service LDAP is a widely spread standard and fits into our requirements of scalability. I find the directory structure to be a good way of separating different layers of information.

LDAP together with X.509 constitute important standards of an authentication service. X.509 can be extended to include access control as in the experiment with RBAC. Trust Management takes a step further and omits the authentication part and focus on access privileges. This is perhaps the best solution in a public network where objects are frequently requested by new and unrecognized subjects.

The prototype fulfilled its requirements as an administrative tool for access control. It was successful because it revealed many new ideas and during the evaluation several

new topics came up. The prototype only covered an administration GUI. Hence, other functions are assumed to manage the access rules correctly.

From the security point of view, I prefer focus on resources instead of users, thus resources are easier to protect that way. I believe that it is convenient to store information about users and resources in distributed directory servers together with corresponding access rules. This is similar to how CAs and repositories work in PKI.

6.2 Conclusions concerning the SMC in LWK

For detailed information of the SMC, see section 4.1 and appendix C.

The current implementation of LWK has the following advantages that can be used as a platform in future projects when fine-granular access control is performed.

- Provides authentication, data integrity and confidentiality.
- Provides all-or-nothing access control. If the user is authorized every request is accepted but if the user is not authorized all requests are denied.
- The SMC constitutes a trusted third party.
- Several SMCs can be connected to form an infrastructure similar to CAs in PKI.
- The possibility to form groups from a set of clients. It makes it easier to scale the system when it grows.
- Auditing functionality.

The following properties have to be added in order to achieve fine-grained access control of the resources.

- A way to separate resources at a server in order to be able to specify fine-grained access rules for them.
- Access control mechanisms either on the server side or on the SMC.
- Administration tool for the access control mechanisms.

6.3 A look into the future

Most probably, the IETF is going to form standards for access control in the future, as they have written standards for authentication up to now. It is a natural order because a user has to be authenticated before access is granted. However, to some extent it is possible that standard organizations do not put much effort into access control, as they may not consider it to be a critical issue.

The following list presents some suggestions for future investigations of access control. They all have to be considered before a system of access control can be implemented.

- How to distribute the access rules to server nodes?
- What is the impact on applications that are using the access control service?
- How do applications have to be adjusted in order to be included in the access control?
- Where should the access control be performed, at the server node or at the SMC?
- How to maintain the access control system?

I recommend the homepages of IETF²¹ and VPNC²² to keep up-to-date with the latest improvements of the standardization work concerning access control in public networks.

²¹ <http://www.ietf.org>

²² <http://www.vpnc.org>

7 Bibliography

- [Alsterlid 1999] Alsterlid, Stefan, *System Specification SMC-MSt*, Sectra internt, 1999.
- [Alsterlid 1999b] Alsterlid, Stefan, *Användarhandledning KNS Systemadministration*, Sectra internt, 1999.
- [Aronius et al 1999] Aronius, Joakim; Wernqvist, Patric, *Public Key Infrastructure Standards and revocation methods*, LiTH-ISY-EX-2035, 1999-02-10
- [Baltimore 2001] Baltimore Technologies Ltd., *Security Considerations for Progressive e-Corporations*, 2001, on the web 2001-08-22 at <http://www.baltimore.com/selectaccess/whitepaper/>
- [Baltimore 2001b] Baltimore Technologies Ltd., *Product Overview*, 2001, on the web 2001-08-22 at <http://www.baltimore.com/selectaccess/whitepaper/>
- [Baltimore 2001c] Baltimore Technologies Ltd., *Product Brochure*, 2001, on the web 2001-08-22 at <http://www.baltimore.com/selectaccess/whitepaper/>
- [Blaze et al 1999] Blaze, M; Feigenbaum J; Ioannidis, J; Keromytis, A, *The role of Trust Management in Distributed Systems Security*, 1999, on the web 2001-11-01 at <http://www.crypto.com/papers/>
- [Blaze et al 2001] Blaze, Matt; Ioannidis, John; Keromytis, Angelos, *Trust Management for IPsec*, 2001, on the web 2001-11-01 at <http://www.crypto.com/papers/>
- [Check Point 1998] Check Point Software Technologies Ltd., *Virtual Private Network Security Components*, 1998, on the web 2001-08-31 at <http://cgi.us.checkpoint.com/rl/resourcelib.asp>
- [Check Point 2000] Check Point Software Technologies Ltd., *FireWall-1 Technical Overview*, 2000, on the web 2001-08-31 at <http://cgi.us.checkpoint.com/rl/resourcelib.asp>
- [Check Point 2001] Check Point Software Technologies Ltd., *VPN-1 Home*, on the web 2001-10-04 at <http://www.checkpoint.com/products/vpn1/>
- [Chen et al 1995] Chen, Fang; Sandhu, Ravi, *Constraints for RBAC*, 1995, on the web 2001-11-06 at http://ite.gmu.edu/list/conference_papers.htm
- [Coulouris et al 2001] Coulouris, George et al, *Distributed Systems, Concepts and design*, Addison-Wesley 2001.
- [Ferraiolo et al 2000] Ferraiolo, David F. et al, *A proposed standard for Role-Based Access Control*, 2000, on the web 2001-10-15 at http://ite.gmu.edu/list/journal_papers1.htm
- [Gollmann 1999] Gollmann, Dieter, *Computer Security*, John Wiley & sons, 1999.
- [Halsey 1996] Halsey, Bill, *Certification Authority FAQ*, 1996, on the web 2001-08-22 at <http://www.anl.gov/ECT/certify/CA-FAQ.html>

-
- [Johansen et al 2000] Johanssen, Sara; Melin, Ulf, *Enhancing an Enterprise Model with Role-Based Access Control Capabilities*, LiTH-IDA-Ex-00/98, 2000
- [Karlsson 1999] Karlsson, Mikael, *SMC System Specification*, Sectra internt, 1999.
- [Kosiur 1998] Kosiur, Dave, *Building and Managing Virtual Private Networks*, John Wiley & sons, 1998.
- [Lucent 2000] Lucent Technologies, *RADIUS white paper*, on the web 2001-09-25 at http://www.livingston.com/marketing/whitepapers/radius_paper.html
- [Malkin 1994] Malkin, G., *The Tao of IET*, 1994, on the web 2001-08-31 at <http://www.ietf.org/rfc/rfc1718.txt>
- [Open Group 1998] Open Group, *DCE Today*, publisher missing, 1998.
- [OPSEC 2000] OPSEC, *Integrated Internet Security Solutions*, on the web 2001-10-04 at <http://www.opsec.com>
- [Organick 1972] Organick E.I., *The Multics System: An Examination of Its Structure*, MIT Press, 1972
- [Parent 2001] Parent, Florent, *Cisco Authentication, Authorization and Accounting Mechanisms*, Syngress Media, excerpts on the web 2001-09-24 at http://www.certmag.com/issues/feb01/dept_excerpts.cfm
- [Park et al 2000] Park, Joon; Sandhu, Ravi; Ahn, Gail-Joon, *Role-Based Access Control on the Web*, 2001, on the web 2001-11-01 at http://ite.gmu.edu/list/journal_papers1.htm
- [Parodi & Burgher] Parodi, John H; Burgher Fred W, *Integrating ObjectBroker and DCE Security*, year missing, on the web 2001-08-22 at <http://www.research.compaq.com/wrl/DECarchives/DTJ/DTJP04/DTJP04SC.TXT>
- [RFC NNNN] Standard documents by IETF, can be found at <http://www.ietf.org/rfc.html> where you just enter the document number denoted by NNNN.
- [RSA 1999] RSA Security Inc., *Understanding Public Key Infrastructure (PKI) – Technology White Paper*, 1999, on the web 2001-09-13 at <http://www.rsasecurity.com/go/Intro-to-PKI/>
- [Sandhu 1997] Sandhu, Ravi S, *Role-Based Access Control*, 1997, on the web 2001-11-05 at <http://www.list.gmu.edu/articles/advcom/a98rbac.pdf>
- [Singh 1999] Singh, Simon, *Kodbooken*, Norstedts Förlag, 1999
- [SSH 2001] SSH, *Introduction to Cryptography*, on the web 2001-10-04 at <http://www.ssh.com/tech/crypto/intro.html>
- [Stallings 1999] Stallings, William, *Cryptography and Network security*, Prentice Hall, 1999.
- [Tanenbaum 1996] Tanenbaum, Andrew S, *Computer Networks*, Prentice Hall, 1996.
- [VPNC 2001] VPNC, *About VPNC*, on the web 2001-10-04 at

-
- [Valencia] <http://www.vpnc.org/>
Valencia, Andrew, *An Overview of the VSTa Microkernel*, on
the web 2001-11-12 at
<http://www.vsta.org/documentation/papers/microkernel.html>
- [Wha] WhatIs.com, *Online dictionary of information technology*, on
the web 2001-10-04 at <http://www.whatis.com>

Appendix A Acronyms

All acronyms listed below are used in the report. They are written out in clear at least on the first occurrence in the text, but except that, only the acronym is used.

ACL	Access Control List
ATM	Automatic Teller Machine
CA	Certification Authority
DAC	Discretionary Access Control
DSA	Digital Signature Algorithm
FTP	File Transfer Protocol
HTTP	Hypertext Transfer Protocol
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
ISAKMP	Internet Security Association Key Management Protocol
ISP	Internet Service Provider
ITU-T	International Telecommunication Union-Telecommunication
KDC	Key Distribution Center
LAN	Local Area Network
LDAP	Lightweight Directory Access Protocol
LWK	LAN/WAN Krypto
MAC	Mandatory Access Control
NSA	National Security Agency
OS	Operating System
PGP	Pretty Good Privacy
PKI	Public Key Infrastructure
POSIX	Portable Operating System Interface for Unix
RADIUS	Remote Authentication Dial-In User Service
RFC	Request For Comment
SA	Security Association
SMC	Security Management Center
SSL	Secure Sockets Layer
TCP/IP	Transmission Control Protocol / Internet Protocol
VPN	Virtual Private Network
WAN	Wide Area Network
WG	Working Group

Appendix B Glossary

Many explanations in this glossary are from IETF's Internet Security Glossary [RFC 2828] and Whatis.com [Wha]. Whatis.com is an online dictionary of information technology.

access control

Protection of system resources against unauthorized access. Access control is a process by which use of system resources is regulated according to a security policy. Access is permitted only to entities (users, programs, processes) that are authorized according to that policy.

accounting

Accounting is a method that records who, what, when and where an action has taken place. (See audit service)

audit service

A security service that records information needed to establish accountability for system events and for the actions of system entities that cause them.

authentication

Ensuring that the data is coming from the source from which it claims to come.

authorization

The action of granting access to a security object.

certificate

See digital certificate.

certification authority (CA)

An entity responsible for establishing and vouching for the authenticity of public keys belonging to users or other CAs. Activities of a CA can include binding public keys to distinguished names through signed certificates, managing certificate serial numbers, and certificate revocation. See section 3.4.1.

client/server relation

Client/server is a relationship between two computer programs in which one program, the client, makes a service request to another program, the server, which fulfils the request. Although programs within a single computer can use the client/server idea, it is a more important idea in a network. In a network, the client/server model provides a convenient way to interconnect programs that are distributed efficiently across different locations. Relative to the Internet, a Web browser can be the client program that requests services, e.g. sending of Web pages, from a Web server somewhere on the Internet. Similarly, a computer with

TCP/IP installed allows you to make client requests for files from FTP (File Transfer Protocol) servers on the Internet.

confidentiality

Preventing unauthorized users from reading or copying data as it travels across a public network.

credentials

Data that is transferred or presented to establish either a claimed identity or the authorizations of a system entity.

data integrity

Ensuring that no one tampers with data as it travels across a public network.

digital certificate

A document constituting a certificate in the form of a digital data object used by a computer) to which is appended a computed digital signature value that depends on the data object. For more information, see section 3.4.1.

digital signature

A digital signature is an electronic signature that can be used to authenticate the identity of the sender of a message or the signer of a document. Furthermore, the signature can be used to ensure that the original content of the message or document is unchanged after it has been sent. See section 3.2.4

directory service

A directory is similar to a database, but typically contains more information that is generally read more often than it is written. Also, directories are designed to contain data that is concise and strictly relevant to the entry. In contrast, databases are designed to hold large amounts of data per entry that may or may not be directly relevant to the entry. For this reason, directories do not usually implement the transaction or rollback schemes that regular databases require. If they are permitted at all, directory updates are typically simple all-or-nothing changes. Directories are tuned to respond quickly to high-volume lookup or search operations.

A lookup is an operation that targets a specific, unique entry, such as a domain name. A search is an operation that targets data common to multiple entries, such as the information collected by an Internet search engine on a topic. Directories may replicate information widely, in order to increase availability and reliability, and thus reduce response time.

extranet

An extranet is a private network that uses the Internet protocol (IP) and the public telecommunication system to securely share parts of a business's information or operations with suppliers, vendors, partners, customers, or other businesses. An extranet can be viewed as part of a company's intranet that is *extended* to users out-

side the company. It has also been described as a “state of mind” in which the Internet is perceived as a way to do business with other companies as well as to sell products to customers.

gateway

A gateway is a network point that acts as an entrance to another network. On the Internet, a node or stopping point can be either a gateway node or a host (end-point) node. Both the computers of Internet users and the computers that serve pages to users are host nodes. The computers that control traffic within the company’s network or at your local Internet service provider (ISP) are gateway nodes.

In the network, a gateway node is often also acting as a proxy server and a firewall server. A gateway is often associated with both a router, which knows where to direct a given packet of data that arrives at the gateway, and a switch, which furnishes the actual path in and out of the gateway for a given packet.

hashing

Hashing is the transformation of a string of characters into a hash value or key that represents the original string. The hash value is of fixed length and usually shorter than the original string. Hashing is used to index and retrieve items, e.g. in a database, because it is faster to find the item using the shorter key than to find it using the original value. It is also used in many signature algorithms. However, the requirements for database hashes and cryptographic hashes are completely different.

The hashing algorithm is also called hash function. In addition to faster data retrieval, hashing is also used for digital signatures. The message is transformed with the hash function, which is then encrypted into a digital signature. Both the message and the signature are sent to the receiver. Using the same hash function as the sender, the receiver derives a message-digest from the message and compares it with the message-digest created by decrypting the signature. They should be the same.

inode

In a Unix-based operating system, an inode is a computer-stored description of an individual file in a Unix file system.

Internet Protocol (IP)

The Internet Protocol is a method or protocol by which data is sent from one computer to another on the Internet. Each computer (known as a host) on the Internet has at least one IP-address that uniquely identifies it from all other computers on the Internet. When sending or receiving data (e.g. an e-mail note or a Web page), the message is divided into little chunks called packets. Each of these packets contains both the sender’s Internet address and the receiver’s address. Any packet is sent first to a gateway computer that understands a small part of the Internet. The gateway computer reads the destination address and forwards the packet to an adjacent gateway that in turn reads the destination address and so forth across the Inter-

net until one gateway recognizes the packet as belonging to a computer within its immediate neighborhood or domain. That gateway forwards the packet directly to the computer whose address is specified.

intranet

An intranet is a private network that is contained within an enterprise. It may consist of many interlinked local area networks and also use leased lines in a WAN (Wide Area Network). Typically, an intranet includes connections through one or more gateway computers to the outside Internet. The main purpose of an intranet is to share company information and computing resources among employees.

ISAKMP

Internet Security Association and Key Management Protocol (ISAKMP) is an Internet IPsec protocol to negotiate, establish, modify, and delete security associations, and to exchange key generation and authentication data, independent of the details of any specific key generation technique, key establishment protocol, encryption algorithm, or authentication mechanism.

ISAKMP provides a framework for authentication and key exchange but does not define them. ISAKMP is designed to be key exchange independent; that is, it is designed to support many different key exchanges.

key escrow

A key recovery technique for storing knowledge of a cryptographic key or parts thereof in the custody of one or more third parties called “escrow agents”, so that the key can be recovered and used in specified circumstances.

man-in-the-middle-attack

A form of active wiretapping attack in which the attacker intercepts and selectively modifies communicated data in order to masquerade as one or more of the entities involved in a communication association.

For example, suppose Alice and Bob try to establish a session key without authentication of the data origin. A “man in the middle” could block direct communication between Alice and Bob and then masquerade as Alice sending data to Bob as well as masquerade as Bob sending data to Alice.

network reference models

Table 6: *The OSI and TCP/IP reference models.*

	OSI	TCP/IP
7	Application	Application
6	Presentation	(not present in the model)
5	Session	(not present in the model)
4	Transport	Transport
3	Network	Internet
2	Data-Link	Host-to-network
1	Physical	

Oakley

Oakley is a key establishment protocol, which describes a series of key-exchanges called “modes”, and details the services provided by each (e.g. identity protection and authentication).

Pretty Good Privacy (PGP)

Pretty Good Privacy (PGP) is a popular program used to encrypt and decrypt e-mail over the Internet. It can also be used to send an encrypted digital signature that lets the receiver verify the sender’s identity and know that the message was not changed while transmitted. Available both as freeware and in a low-cost commercial version, PGP is the most widely used privacy-ensuring program by individuals and is also used by many corporations. PGP has become a de facto standard for e-mail security.

protocol

A set of rules (i.e. formats and procedures) to implement and control some type of association (e.g. communication) between systems. In particular, a series of ordered steps involving computing and communication that are performed by two or more system entities to achieve a joint objective.

proxy server

In a computer system with connection to the Internet, a proxy server is a server that acts as an intermediary between a workstation user and the Internet so that the system can ensure security, administrative control, and possibly caching service. A proxy server is associated with or part of a gateway server that separates the inner network from the outside network and a firewall server that protects the inner network from outside intrusion.

A proxy server receives a request for an Internet service (such as a Web page request) from a user. If it passes filtering mechanisms, the proxy server (assuming it is also a cache server) looks in its local cache for previously downloaded Web pages. If it finds the page, it returns it to the user without needing to forward the request to the Internet. If the page is not in the cache, the proxy server, acting as a

client on behalf of the user, uses one of its own IP addresses to request the page from the server out on the Internet. When the page is returned, the proxy server relates it to the original request and forwards it on to the user.

reference monitor

An access control concept that refers to an abstract machine that mediates all accesses to objects by subjects.

router

On the Internet, a router is a device or, in some cases, software in a computer, that determines the next network point to which a packet should be forwarded towards its destination. The router is connected to at least two networks and decides which way to send each information packet based on its current understanding of the state of the networks it is connected to. A router is located at any gateway where one network meets another. A router is often included as a part of a network switch.

RSA (Rivest, Shamir, and Adleman)

RSA is an encryption and authentication system that uses an algorithm developed in 1977 by Ron Rivest, Adi Shamir, and Leonard Adleman. The RSA algorithm is among the most commonly used encryption and authentication algorithm and is included as a part of the Web browsers from Netscape and Microsoft. RSA Security owns the encryption system. The company licenses the algorithm technologies and also sells development kits. The technologies are part of existing or proposed Web, Internet, and computing standards.

Security Association

Security Association (SA) is concept used in IPsec, which describes the relationship established between two or more entities to enable them to protect data they exchange. The relationship is used to negotiate characteristics of protection mechanisms, but does not include the mechanisms themselves.

A SA describes how entities will use security services. The relationship is represented by a set of information that is shared between the entities and is agreed upon and considered a contract between them.

Security Policy

A set of rules and practices that specify or regulate how a system or organization provides security services to protect sensitive and critical system resources.

An *identity-based* security policy is based on the identities and/or attributes of users, a group of users, or entities acting on behalf of the users and the resources/objects being accessed.

On the other hand, a *rule-based* security policy is based on global rules imposed for all users. These rules usually rely on comparison of the sensitivity of the resource being accessed and the possession of corresponding attributes of users, a group of users, or entities acting on behalf of users.

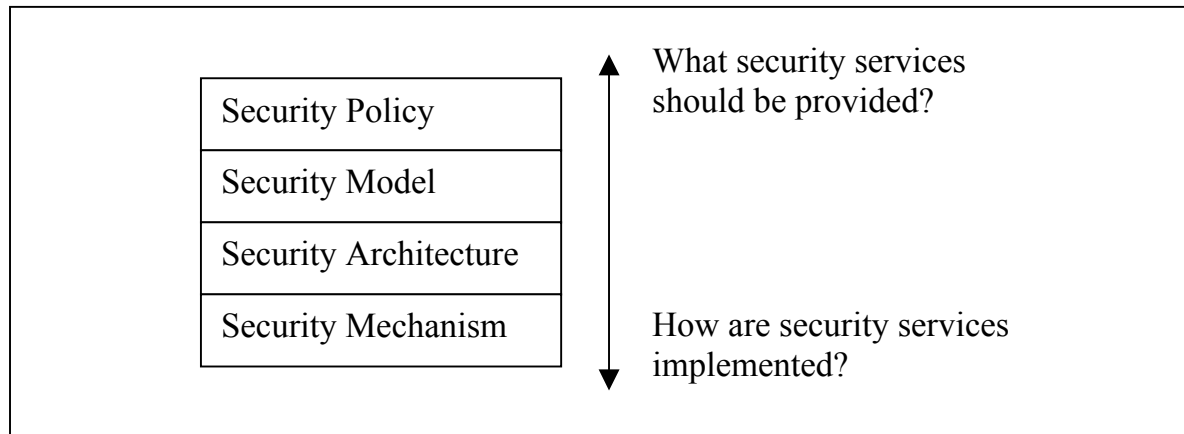


Figure 18: *Four layers of security engineering.*

The security policy is one of four layers of the security engineering process (as shown in Figure 18). Each layer provides a different view of security, ranging from what services are needed to how services are implemented.

TCP/IP

TCP/IP (Transmission Control Protocol / Internet Protocol) is the basic communication language or protocol of the Internet. It can also be used as a communications protocol in a private network (either an intranet or an extranet). Computers with direct access to the Internet are provided with a copy of the TCP/IP program just as every other computer that you may send messages to or get information from.

TCP/IP is a two-layer program. The higher layer, TCP, manages the assembling of a message or file into smaller packets that are transmitted over the Internet and received by a TCP layer that reassembles the packets into the original message. The lower layer is the IP layer and is described separately. TCP/IP uses the client/server model of communication in which a computer user (a client) requests and is provided a service (such as sending a Web page) by another computer (a server) in the network. TCP/IP communication is primarily point-to-point, meaning each communication is from one point (or host computer) in the network to another point or host computer.

Appendix C Security Management Center (SMC)

The objective of this appendix is to explain how SMC (Security Management Center) in LWK works. The SMC system is used together with both a secure GSM system and the LWK system. The SMC is a common part for the two systems because it solves a task that is shared by the both systems. However, in this appendix, only the LWK-part of the SMC is covered. The sources for this appendix are [Karlsson 1999], [Alsterlid 1999], and [Alsterlid 1999b].

C.1 Purpose

The main purpose of the SMC is to generate and distribute keys and tickets to the clients. This is done on requests from the clients. The major services of the SMC are:

- Generation of session tickets on client requests.
- Generation of multi-user keys and distribution on client request (optional).
- Directory services (optional).
- Communication with cooperating SMCs.
- Distribution of alarms to specific alarm receivers.

Then last two services are typical for the LWK system.

C.2 Environment of the SMC

An administrator manages the SMC with the local management console. The SMC connects to its users (clients) and master SMC through external network connections. Communication on the external network is based on IP. The SMC environment is depicted in Figure 19.

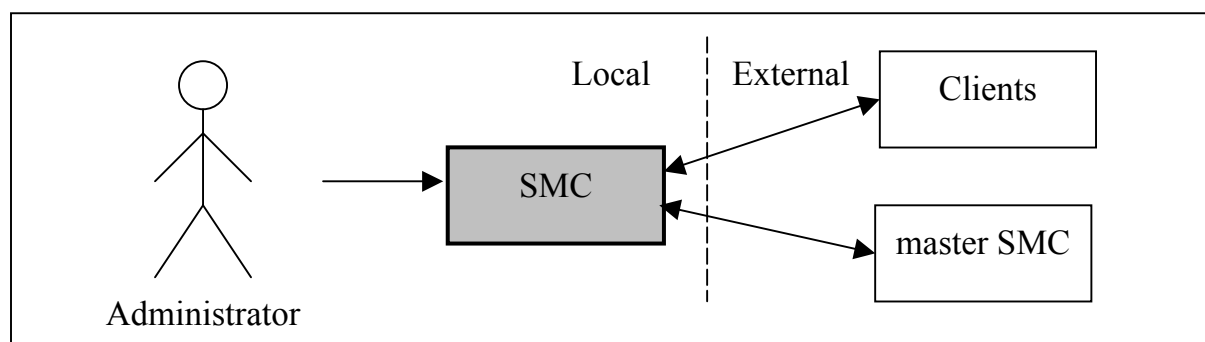


Figure 19: *Environment of the SMC.*

C.3 Design

The SMC consists of three subsystems connected by two networks (see Figure 20). We are only going to look at two of the subsystem (MSt and KS) because the third (GW) is not an issue for access control.

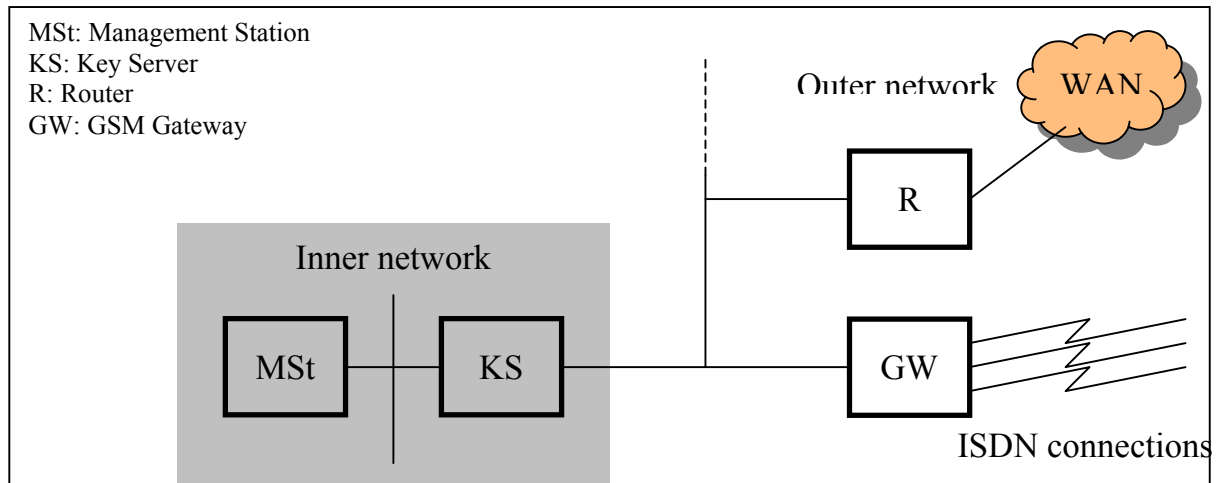


Figure 20: *Overall structure of the SMC.*

The management station (MSt) and key server (KS) are connected via the inner network. Together they handle and store all security relevant data in the system, such as keys, access tables and catalogues. They shall therefore be considered sensitive/secret and kept in a secure location as indicated by the gray field in the figure. All the clients attach to the SMC via the outer network.

C.3.1 Management Station

The management station (MSt) contains user interfaces for configuration and management of the SMC. Furthermore, the MSt receives and stores log messages from the key server. The communication between the MSt and the key server goes over the inner network and uses the key server management protocol. The management station is implemented as software running on a PC under Windows NT 4.0.

The MSt is positioned in a physical secure area together with the key server. The MSt is used by the administrator to configure the key server and its database containing information about clients, domains etc.

Functional Behavior. Configuration and management of the key server and other administrative tasks are performed on the management station. The MSt provides the following services.

- Setting of clients, client groups and access rules in the key server database.
- Backup of the KS database.
- Key management.
- Loading of keys. Keys are either read from diskette by the MSt, or from bar code or smart card by the key server.
- Local configuration of the key server. This includes things like setting of IP-addresses.
- Testing of the key server hardware.
- Access to and management of the key server logs.

Cooperation with the key server. The key server contains three databases, one with administrators, one with keys and one with clients, groups and access rules. The key and administrator databases always reside in the key server. The MSt can, however, give commands that affect these databases. These commands are for loading new keys etc. The other database also always resides in the key server, but it is managed in the MSt. When the MSt is started, this database is copied to the MSt. If the administrator makes any changes in the database it has to be copied back to the key server, in order for the changes to have effect.

C.3.2 Key Server

The key server (KS) contains all cryptographic functions and manages all plaintext keys. The KS also stores all the databases needed to perform access control and session ticket generation. The KS communicates with the MSt via the inner network and with its clients via the outer network. The key server is the only connection between the inner and the outer network and is responsible for the separation of the two sides.

The key server uses public-key algorithms to authenticate administrators of the SMC but symmetric algorithms to authenticate clients. Dedicated hardware is used to implement the key server.

C.4 Internal System Interfaces

In this section we take a closer look at databases and one protocol that are part of the internal structure of the SMC.

C.4.1 SMC Databases

Information about clients and access rules are stored in a database in the key server. The key server uses this information to determine how to answer requests for session tickets. The SMC database information is communicated between the management station and the key server using the key server management protocol.

The SMC database is structured into clients, client groups, and rules.

LAN/WAN Clients. A LAN/WAN client is an application (or group of applications) running on a host with LAN/WAN software and hardware installed. The following information is stored about LAN/WAN clients in the SMC.

- Name. Unique identifier within this SMC system (domain).
- Name of the domain to which the client belong.
- Name of the system to which the client belong.
- Current master key.
- Master key for next period.
- Zero or more “aliases”.
- Logging rules.

An alias can be either an Internet style email address (RFC 822), a DNS-name or some free-form text. Aliases are the type of client identifier that a user is most likely to provide at a client. Logging rules are used to determine what logging (if any) shall be performed for accesses to this client.

Client Groups. A client group consists of zero or more other client groups and clients. A client group can be seen as a convenient way of selecting several clients at the same time. For every client group the following information is stored.

- Group name.
- Names of all member clients.
- Names of all member client groups.

Note that both clients and client groups can be members of several client groups at the same time.

Access rules. Access rules are used to express who is allowed to communicate with whom. Hence, they do not specify any details of what is accessed on the object side. The access rules are based on the unique names given to each client in the system. The only parameters of an access rule, except identifiers for subject and object (client A and client B), are restrictions on the generated session key.

The following information is stored about LAN/WAN access rules:

- A client identifier. Can be either a client name or the name of a group.
- B client identifier. Can be either a client name or the name of a group.
- Usage information. Specifies the maximal amount of data to be encrypted by a session key generated from this rule.
- Parameters for generating session keys. Lifetime of generated keys etc.
- Information about which logging (if any) that should be done for this rule.

LAN/WAN access rules are not automatically symmetrical. That is: client A may contact client B but not the other way around.

C.4.2 Key Server Management Protocol

The key server management protocol is used on the inner network of the SMC. It is used to communicate user management actions and configuration data from the management station to the key server.

The key server management protocol is used only within the protected area of the SMC and is not encrypted or otherwise protected against malicious actions.

C.5 External System Interfaces

In this section we gleam at a protocol that is a part of the SMC's external system interfaces.

C.5.1 Key Server Access Protocol

The key server access protocol is used on the outer network of the SMC. LAN/WAN clients and the GSM gateway use the protocol to communicate session tickets and management data with the key server. It is also used for communication between co-operating key servers.

The key server access protocol is encrypted and provides confidentiality, integrity and authentication.

“The nice thing about standards is that there are so many to choose from.”

Andrew S. Tanenbaum

Appendix D Standardization

The first part of this appendix describes different standardization organizations. In the second part, some standards related to access control are given. Most of the standards treat authentication issues.

D.1 Standardization Organizations

The next sections introduce a number of organizations that are more or less involved in the access control area.

D.1.1 Internet Engineering Task Force (IETF)

The Internet Engineering Task Force (IETF) is a large open international community of network designers, operators, vendors, and researchers concerned with the evolution of the Internet architecture and the smooth operation of the Internet. It is open to any interested individual. It is the principal body engaged in the development of the new Internet standard specifications.

The actual technical work of the IETF is done in its working groups (WGs), which are organized by topic into several areas (e.g. routing, transport, security, etc.). Much of the work is handled via mailing lists. RFC (Request For Comments) is the humble name for documents produced by the working groups. The IETF holds meetings three times per year.

The IETF meetings are not conferences, although there are technical presentations. The IETF is not a traditional standards organization, although many specifications are produced that become standards. The IETF is made up of volunteers who meet three times a year to fulfill the IETF mission.

There is no membership in the IETF. Anyone may register for and attend any meeting. The closest thing there is to being an IETF member is being on the IETF working group mailing lists. The mailing lists contain the best information about current IETF activities. [Malkin 1994]

AAC (Authorization and Access Control) Working Group

This working group was concluded in March 1995. However, the purpose of the WG is still interesting to examine because it almost perfectly covers the scope of this thesis. I have not found any reason to why it was concluded because the WG's goal is still not fulfilled. I have tried to reach IETF to ask for the reason but without success. Perhaps the reason to why no RFCs were produced was that supporting standards at that time were mature enough.

The goal of the Authorization and Access Control working group is to develop guidelines and an API through which network accessible applications can uniformly specify access control information. This API will allow applications to make access control decisions when clients are not local users, might not be members of a common organization, and often not known to the service or application in advance.

Several authentication mechanisms are in place on the Internet, but most applications are written with local application and access control based on the output of such authentication mechanisms.

A second, longer-term purpose of the working group is to examine evolving mechanisms and architectures for authorization in distributed systems and to establish criteria, which enable internetworking of confidence and trust across systems.

IPsec (Internet Protocol Security) Working Group

The IPsec (Internet Protocol Security) working group develops standards for security at the network or packet-processing layer of network communication. Earlier security approaches have inserted security at the application layer of the communication model. IPsec will be especially useful for implementing VPNs and for remote user access through dial-up connection to private networks. A big advantage of IPsec is that security arrangements can be handled without requiring changes to individual user computers. Cisco has been a leader in proposing IPsec as a standard (or combination of standards and technologies) and has included support for it in its network routers. [whatis.com]

IPSP (IP Security Policy) Working Group

The rapid growth of the Internet and the need to control access to network resources (bandwidth, routers, hosts etc.) has quickly generated the need for representing and managing the policies that control access to these resources in a scalable, secured and reliable fashion.

Current IP security protocols and algorithms can exchange keying material using IKE and protect data flows using the AH and/or ESP protocols. The scope of IKE limits the protocol to the authenticated exchange of keying materials and associated policy information between the end-points of a security association.

However, along the path of communication, there may be administrative entities that need to impose policy constraints on entities such as security gateways and router filters. There also is a need for endpoints of a security association and/or, for their respective administrative entities, to securely discover and negotiate access control information for the end hosts and for the policy enforcement points (security gateways, routers, etc.) along the path of the communication.

To address these problems the IPSP Working Group will among other things:

- Develop or adopt an extensible policy specification language. The language should be generic enough to support policies in other protocol domains, but must provide the necessary security mechanisms that are vital to IPsec.
- Provide guidelines for the provisioning of IPsec policies using existing policy distribution protocols. This includes profiles for distributing IPsec policies over protocols such as LDAP and FTP.
- Adopt or develop a policy exchange and negotiation protocol.

This group will also coordinate with other IETF WGs on specifying policies and policy schemas in order to maintain compatibility and interoperability. In particular, this working group will work closely with the Policy Framework WG to ensure that the IPsec Policy Information and data model fits and can be supported within the general Policy Framework.²³

IPSRA (IP Security Remote Access) Working Group

The goals of this working group are:

- To define a remote access architecture. The entities participating in the remote access and their relationships will be defined in a framework document.
- To define a standard mechanism to accomplish human user authentication to an IPsec device running IKE, using legacy authentication mechanisms.

The working group strongly prefers mechanisms that require no changes to AH, ESP or IKE protocols. For an explanation of AH, ESP, please see section D.2.2.

PKIX (Public-Key Infrastructure X.509) Working Group

The PKIX working group was established in the fall of 1995 with the intent of developing Internet standards needed to support an X.509-based PKI. Several information and standard track documents in support of the original goals of the WG have been approved by the IESG. The standards, among other things, profiles the X.509 version 3 certificates and version 2 CRLs for use in the Internet. Work continues on a second certificate management protocol, CMC, closely aligned with the PKCS publications and with the cryptographic message syntax (CMS). A roadmap, providing a guide to the growing set of PKIX documents, is also being developed as an informational RFC.

The working group is now embarking on additional standards work to develop protocols that are either integral to PKI management, or that are otherwise closely related to PKI use. Work is ongoing on alternative certificate revocation methods.

Policy Framework Working Group

There is a need to represent, manage, share, and reuse policies and policy information in a vendor-independent, interoperable, and scalable manner. This working group has three main goals. Firstly, to provide a framework that will meet these needs. Secondly,

²³ The working group has up to now produced four Internet-Drafts and zero RFCs.

to define an extensible information model and specific schemata compliant with that framework that can be used for general policy representation (called the core information model and schema). For now, only a directory schema will be defined. Thirdly, to extend the core information model and schema to address the needs of QoS traffic management.²⁴

D.1.2 VPNC

VPNC stands for Virtual Private Network Consortium and is an organization working for interoperability among VPN products. Members of VPNC span the range from large to small manufacturers of VPN hardware and software. Among the members are Microsoft, Cisco, RSA Technology, Baltimore Technologies, and Check Point Software.

In order to determine if a product is VPN-interoperable VPNC has developed a conformance test. The primary goals of VPNC are:

- Promote the products of its members to the press and to potential customers.
- Increase interoperability between members showing where the products interoperate.
- Server as the forum for the VPN manufacturers throughout the world.
- Help the press and potential customers understand VPN technologies and standards.
- Provide publicity and support for interoperability testing events.

It should be noted that VPNC do not create standards; instead, it strongly supports the current and future IETF standards. [VPNC 2001]

D.1.3 OPSEC

The OPSEC (Open Platform for Security) Alliance was founded in 1997 by Check Point Software²⁵ to provide complete, integrated multi-vendor security solutions. Using the OPSEC software development kit (SDK) and industry standards, vendors have been developing products over the last three years, delivering complimentary security applications defined and driven by a single, central, enterprise-wide security policy.

Today the OPSEC platform boasts the broadest operating system and network infrastructure support and its Internet security integration interfaces have been adopted by more third parties than any other security platform in the industry. With over 150 partners, OPSEC try to offer customers the broadest choice of integrated applications and services that support Check Point's Secure Virtual Network Architecture. [OPSEC 2000]

²⁴ Up to now the working group has produced five Internet-Drafts and one RFC.

²⁵ Check Point has developed VPN-I, which is described in Appendix E.

D.1.4 Open Group

The Open Group is a software standards organization that is sponsored by a number of major software vendors. The Open Group develops and fosters industry standards for software interfaces, often using technologies developed by one of the sponsoring companies. The Open Group originated by combining two previous organizations, X/Open and the Open Software Foundation (OSF). Standards that the Open Group maintains include the standard Unix program interfaces and LDAP.²⁶ Open Group is also developing a product, DCE, which is described in Appendix E.

D.1.5 IEEE

The IEEE (Institute of electrical and Electronics Engineers) describes itself as “the world’s largest technical professional society – promoting the development and application of electro technology and allied sciences for the benefit of humanity, the advancement of the profession, and the well-being of our members.”

The IEEE fosters the development of standards that often become national and international standards. The organization publishes a number of journals, has many local chapters, and several large societies in special areas, such as the IEEE Computer Society.²⁷ From our point of view, IEEE's most interesting standard series is POSIX 1003.6.

D.2 Standards and Products

The following sections describe a number of standardized products that involve access control in one way or another.

D.2.1 IKE (Internet Key Exchange)

IKE is the key management protocol used in IPsec. It combine two protocols: ISAKMP and Oakley. IKE is used to dynamically create encryption keys and Security Associations. It provides four capabilities:

- Agreement between parties on proper protocols, algorithms, and keys
- Authentication of communicating parties
- Secure exchange of keys
- Management of keys after they have been exchanged

There are modes in IKE, two of which are initial communications used to define secure communications for future key exchanges (Phase 1) and one that is used for subsequent key exchanges (Phase 2).

- *Main Mode* – A six step (three round trips) Phase 1 exchanges that protects the identities of both communicating parties. Used to generate the security rules for a Phase 2 exchange.

²⁶ For more information, see <http://www.opengroup.org>

²⁷ For more information, see <http://www.ieee.org>

- *Aggressive Mode* – A three step Phase 1 exchange that is quicker than Main Mode, but does not protect the identities of the communicating parties. Used to generate the security rules for a Phase 2 exchange.
- *Quick Mode* – A three step Phase 2 exchange that is used to generate the security rules (Security Association) for general IPsec communications and creating new keys when needed.

IKE is an industry VPN standard protocol that simplifies key management by automating the key exchange process. This automation has allowed the development of large-scale extranet projects, such as the Automotive Network Exchange (ANX).

D.2.2 IETF IPsec

As mentioned earlier, IETF has a working group called IP Security (IPsec). One of the most important aspects of the IPsec WG is the IPsec standard itself, which is defining the overall IP packet structure and security associations relative to VPN communications. There are two major concepts in IPsec: Authentication Header and Encapsulated Security Payload. They are defined as follows:

- Authentication Header (AH) is used to provide connectionless integrity and data origin authentication for an entire IP datagram (hereafter referred to as “authentication”).
- Encapsulating Security Payload (ESP) provides authentication and encryption for IP datagrams with the encryption algorithm determined by the user. In ESP authentication, the actual message digest is now inserted at the end of the packet (whereas in AH the message digest is inside the packet).

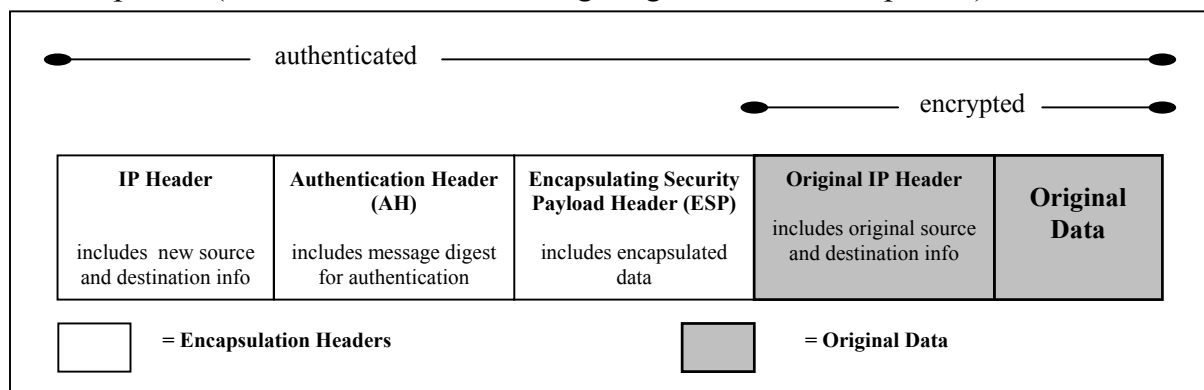


Figure 21: IPsec packet structure.

AH provides data integrity only and ESP, formerly encryption only, now provides both encryption and data integrity. The difference between AH and ESP data integrity is the scope of the data being authenticated. AH authenticates the entire packet, while ESP does not authenticate the outer IP header. In ESP authentication, the actual message digest is now inserted at the end of the packet, whereas in AH the digest is inside the authentication header, as shown in Figure X above.

The IPsec standard dictates that prior to any data transfer occurring, a Security Association (SA) must be negotiated between the two VPN nodes (gateways or clients).

The SA contains all the information required for execution of various network security services such as the IP layer services (header authentication and payload encapsulation), transport or application layer services, and self-protection of negotiation traffic. These formats provide a consistent framework for transferring key and authentication data which is independent of the key generation technique, encryption algorithm and authentication mechanism.

One of the major benefits of the IPsec efforts is that interoperability with third party VPN solutions is facilitated. However, it does not provide an automatic mechanism to exchange the encryption and data authentication keys needed to establish the encrypted session, which introduces the second major benefit of the IPsec standard: key management infrastructure or PKI.

IPsec PKI and Key Management

Part of the IETF IPsec standard is the definition of an automated key management scheme, which includes the concept of a Public Key Infrastructure (PKI), which is an open community of Certificate Authorities, which in most cases uses a hierarchical model to construct trust associations where none had existed before. A PKI is important when setting up a VPN between a corporate network and a business partner or supplier because it requires a secure key exchange from a third party CA that is trusted by both VPN nodes. The mandatory automated key management scheme defined by IETF IPsec for IPv6 is ISAKMP/Oakley (Internet Security Association and Key Management) with SKIP (Simple Key management for IP) defined as optional.

D.2.3 Kerberos

Kerberos is a secure method for authenticating a request for a service in a computer network. Kerberos was originally developed at the Massachusetts Institute of Technology (MIT). Later, improvements have been done by the IETF in [RFC 1510]. The name is taken from Greek mythology; Kerberos was a three-headed dog that guarded the gates of Hades. Kerberos lets a user request an encrypted “ticket” from an authentication process that can then be used to request a service from a server. The user’s password does not have to pass through the network. A trial version of Kerberos (client and server) can be downloaded from MIT or you can buy a commercial version.

Briefly and approximately, here is how Kerberos works:

1. Suppose you want to access a server on another computer (which you may get to by sending a Telnet or similar login request). You know that this server requires a Kerberos “ticket” before it will honor your request.
2. To get your ticket, you first request authentication from the Authentication Server (AS). The AS creates a “session key” (which is also an encryption key) basing it on your password (which it can get from your user name) and a random value that represents the requested service. The session key is effectively a “ticket-granting ticket”.
3. Next, you send your ticket-granting ticket to a ticket-granting server (TGS). The TGS may be physically the same server as the AS, but it is now performing

a different service. The TGS returns the ticket that can be sent to the server for the requested service.

4. The service either rejects the ticket or accepts it and performs the service.
5. Because the ticket you received from the TGS is time-stamped, it allows you to make additional requests using the same ticket within a certain time period (typically, eight hours) without having to be reauthenticated. Making the ticket valid for a limited time period make it less likely that someone else will be able to use it later.

The actual process is much more complicated than just described. The user procedure may vary somewhat according to implementation.

D.2.4 Secure Sockets Layer / Transport Layer Security (SSL/TLS)

The Secure Sockets Layer (SSL) is a commonly used protocol for setting up an encrypted tunnel through a public network. SSL has recently been succeeded by Transport Layer Security (TLS), which is based on SSL. SSL uses a program layer located between the Internet's HTTP and TCP layers. SSL is included as part of both the Netscape and Microsoft browsers and most Web server products. Developed by Netscape, SSL also gained the support of Microsoft and other Internet client/server developers as well and become the de facto standard until evolving into Transport Layer Security. The "sockets" part of the term refers to the sockets method of passing data back and forth between a client and a server in a network or between program layers in the same computer. SSL uses the public-and-private key encryption system from RSA, which also includes the use of a digital certificate.

TLS and SSL are an integral part of most Web browsers (clients) and Web servers. If a Web site is on a server that supports SSL, SSL can be enabled and specific Web pages can be identified as requiring SSL access.

TSL and SSL are not interoperable. However, a client that handles SSL but not TLS can handle a message sent with TLS.

D.2.5 POSIX

POSIX (Portable Operating System Interface for Unix) is a set of standard operating system interfaces based on the Unix operating system. They include mechanisms for access control by ACLs and capabilities. The need for standardization arose because enterprises using computers wanted to be able to develop programs that could be moved among different manufacturers' computer systems without having to be re-coded. Unix was selected as the basis for a standard system interface partly because it was "manufacturer-neutral". However, several major versions of Unix existed so there was a need to develop a common denominator system.

POSIX.1 and POSIX.2 interfaces are included in a somewhat larger interface known as the X/Open Programming Guide 4.2.

The POSIX interfaces were developed with guidance of the IEEE. The document of POSIX 1003.6, which contains specifications for access control, is not freely available but is sold by IEEE.

D.2.6 Digital Signature Standard (DSS)

Digital Signature Standard (DSS) is the digital signature algorithm (DSA) developed by the U.S. National Security Agency (NSA) to generate a digital signature for the authentication of electronic documents. DSS was put forth by National Institute of Standards and Technology in 1994, and has become the U.S. government standard for authentication of electronic documents. DSS is specified in Federal Information Processing Standard 186. [Wha]

D.2.7 Public-Key Cryptography Standards (PKCS)

The Public-Key Cryptography Standards (PKCS) are a set of inter-vendor standard protocols for making possible secure information exchange on the Internet using a public key infrastructure (PKI). The standards include RSA encryption, password-based encryption, extended certificate syntax, and cryptographic message syntax for S/MIME, RSA's proposed standard for secure e-mail. RSA Laboratories in cooperation with a consortium that include Apple, Microsoft, DEC, Lotus, Sun, and MIT developed the standards. [Wha]

D.2.8 RADIUS

RADIUS (Remote Authentication Dial-In User Service) is a client/server protocol and software that enables remote access servers to communicate with a central server to authenticate dial-in users and authorize their access to the requested system or service. RADIUS allows an organization to maintain user profiles in a central database that all remote servers can share. It provides security, allowing a company to set up a policy that can be applied at a single administrated network point. Having a central service also means that it is easier to track usage for billing and for keeping network statistics. Created by Livingston (now owned by Lucent), RADIUS is a de facto industry standard used by Ascend and other network product companies and is a proposed IETF standard. [RFC 2058]

RADIUS is a system that secures private networks against unauthorized access. RADIUS includes two pieces: an authentication server and client protocols. The server is installed on a central computer in the private network. RADIUS is designed to simplify the security process by separating security technology from communications technology. All user authentication and network service access information is located on the authentication, or RADIUS, server. This information is contained in a variety of formats suitable to the customer's requirements. RADIUS in its generic form will authenticate users against a Unix password file, Network Information Service (NIS), as well as a separately maintained RADIUS database. Communication servers operate as RADIUS clients. The RADIUS client sends authentication requests to the RADIUS server and acts on responses sent back by the server. [Lucent 2000]

Appendix E External Products

In this appendix three external products are introduced. They all involve mechanisms for access control but they are implemented differently.

E.1 DCE by Open Group

According to [Wha], DCE (Distributed Computing Environment) is an industry-standard software technology for setting up and managing computing and data exchange in a system of distributed computers. DCE is typically used in a larger network that include different size servers scattered geographically. DCE uses the client/server model. Using DCE, application users can use applications and data at remote servers. Application programmers need not to be aware of where their programs will run or where the data will be located.

Much of DCE setup requires preparation of distributed directories so that the DCE application and related data can be located when they are being used. DCE includes security support and some implementations providing support for access to databases.

DCE was developed by the Open Software Foundation (OSF), which nowadays is incorporated into The Open Group.

E.1.1 Privilege Server

The DCE Security Service provides the mechanism for global identity. The mechanism is based on Kerberos²⁸ authentication, which is a private or symmetric key scheme (as opposed to a public or asymmetric key scheme). A private key scheme requires some trusted third-party node to act as a distribution center for encryption key or credentials. Each node or user has a key that is known only to the user and the distribution center. In DCE Security, the distribution center is known as a privilege server.

The following is a simplified description of the encryption key protocol between the privilege server and a client. The actual key exchange protocol, which uses three exchanges and conversion keys, results in a Privilege Access Certificate (PAC) in the possession of a client. The PAC, which is appended to each request, contains the authorization information to be compared with the access control information stored with the application server.

When a client wishes to communicate with a server, each must acquire a time-stamped session key for secure communication. The session key is protected in several ways. The time stamp means that the key is only valid for a limited time, which protects against brute-force-attempts to break the key and reuse it. Also, each key is host-

²⁸ Described in Appendix D.2.3

specific and can only be used from the node for which it is issued. Finally, the session key is never sent over the network in unencrypted form. [Parodi & Burgher]

E.1.2 Access Control Mechanisms

In DCE, the information required to make the authorization decision is contained in an Access Control List (ACL). The ACL contains a list of entries describing either explicit users or groups of users and a set of access permissions for each. By allowing users to be organized in groups having the same access rights and requirements, DCE makes administration of a complete information environment much easier, especially when scaled to very large sizes. New users can simply be put into appropriate groups, minimizing the need to set individual ACLs for specific users (although explicit per-user ACL entries can always be made for exceptional situations.)

DCE access control lists are a superset of the access lists specified in the POSIX 1003.6 standard. POSIX ACLs were designed to control the file access of users sharing a single computer. The extension of DCE ACLs allows them to be useful in a distributed computing environment.

DCE allows lists of users to have ownership and administrative rights to distributed resources. This allows more than one individual to control the types of access other users may have to these resources. In addition, DCE allows ACLs to contain names of individuals from organizations other than the one that administers the resources the ACLs protect. For instance, a user from a company in Boston can be given individual access to a laser printer at a different company in Munich. POSIX ACLs, on the other hand, limit inclusion to members within one organization, and are applicable only to files. [Open Group 1998, p149]

Components of an ACL

In DCE an ACL consists of the following [Open Group 1998]:

- An *ACL manager* type identifier, which identifies the manager type of the ACL.
- A default cell identifier, which specifies the cell of which a principal or group identified as local is assumed to be a member.
- At least one ACL entry.

The *ACL manager* is that portion of a server that handles ACLs. One ACL manager can support several different types of ACLs. From a more abstract point of view, a corresponding ACL manager type supports each ACL type.

When a principal requests access to a DCE object associated with an ACL, the object's ACL manager compares the UUIDs of the principal and any groups of which the principal is a member (the principal's privilege attributes) with the UUIDs of the principals and groups listed in the ACL entry. It does this simply by reading through the list of ACL entries. The manager grants the access permissions in the first ACL entry (or entries in the case of groups) it finds that match any of the principal's privilege attrib-

utes. If the permissions in the matching entry allow the requested mode of access, the principal gains access; if not, access is denied.

The ACL authorization mechanism is used to control access to DCE objects. ACLs are associated with files, directories, CDS entries, and registry objects. They can be implemented also by arbitrary applications to control access to their internal data objects. Each ACL consists of multiple ACL entries that define who is authorized to do what to the object, specifically [Open Group 1998]:

- Who can access the object.
- What kinds of access those principals or groups have to the object.
- What kind of access is allowed to unauthenticated users.

Please note that in this paragraph the term *user* is analogous to principal. A principal can be a human user, server, or a machine.

Sparse ACLs

The authorization model implemented by WebCrusader extends the standard DCE ACL model by the use of sparse ACLs. In a sparse ACL model not every object in the hierarchy has to have an explicit ACL. If an object does not have a real ACL it is said to have an inherited ACL; that is, it inherits the ACL of the next superior object in the hierarchy that has a real ACL. Many ACL models include the idea of inheriting a default ACL at the time the ACL is created, but once ACLs exist they don't change even if ACLs superior to them do. With a sparse ACL model, ACLs must be created only at those relatively few points in the hierarchy where access control differs. This can significantly reduce the number of ACLs the administrator must track and maintain, which makes the authorization model much more manageable, especially for large and complex hierarchies of objects. [Open Group 1998, p156]

Authorization

DCE supports two authorization mechanisms: named-based and ACL authorization. The client specifies which of these two mechanisms to use when the client uses authenticated RPC or GSS to authenticate to the server. In our case the ACL mechanism seem most appropriate because of the following advantages: [Open Group 1998, p215]

- Granular configuration
- Considered superior by many security authorities

If the client specifies ACL-based authorization when authenticating to the server, the client obtains credentials to the server containing the client's Privilege Attribute Certificate (PAC) and passes these credentials to the server. PAC contains the client's user identity as well as the identity of each group of which the client is a member. Then, when the client requests access to an object protected by the server, the server determines whether the client is authorized by obtaining the client's user and group identities from the client PAC and comparing with the identities and permissions that are listed in an ACL previously configured for the object.

An example of ACL in DCE

For example, assume Alice uses GSS-API (a security interface) to authenticate with the payroll server and specifies ACL-based authorization. Using GSS-API, Alice obtains credentials containing Alice's PAC and then passes these credentials, encapsulated in a GSS-API token, to the payroll server. The PAC contains Alice's identity and the identity of each group of which Alice is a member. Then, assume that Alice requests to read Bob's payroll record and an administrator previously configured an ACL for Bob's record. The payroll server retrieves Alice's PAC from the credentials; then determining whether Alice is authorized to access Bob's payroll record by comparing the identities in Alice's PAC with the list of identities and permissions in the ACL that are allowed for each identity. If the ACL reveals that any of the identities in Alice's PAC are allowed read permission on Bob's payroll records, the server determines that Alice is authorized to read Bob's payroll record. [Open Group 1998, p214]

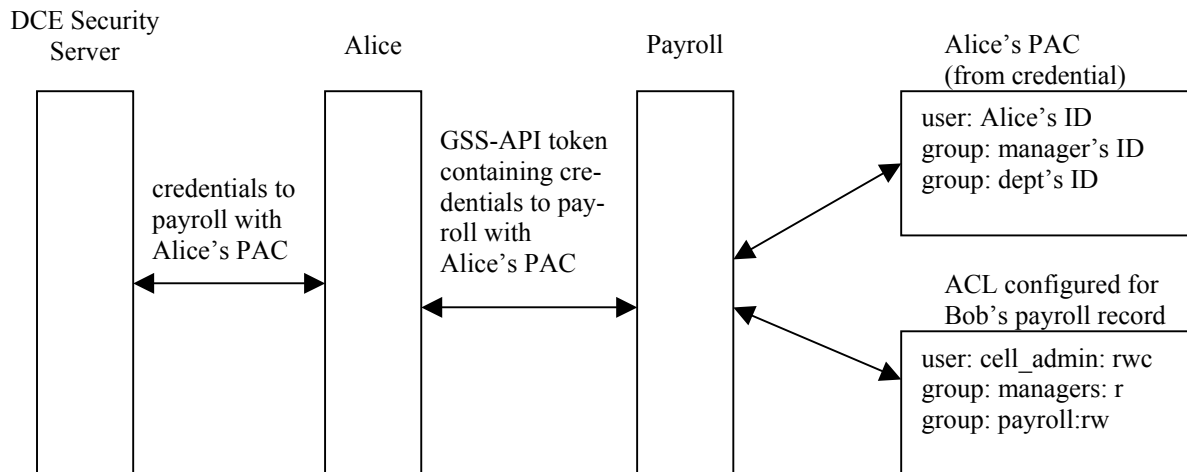


Figure 22: *Example of ACL-based authorization.*

File systems are frequently designed to provide access permissions for file system objects, such as files and directories. ACLs in DCE are more extensive. In DCE, many objects can have ACLs and be assigned permissions. DCE ACLs control access to objects managed by DCE components, like the Distributed File Service, the DCE Security Service, and the DCE Directory Service.

ACLs for the security service (the component that controls accounts) can, for example, authorize certain principals to change all of the information associated with an account, authorize other principals to change only a subset of the information associated with accounts, restrict other principals from changing any of the information associated with accounts.

DCE can support particular sets of permissions that correspond to particular types of objects. This extensive usage of ACLs is in contrast to that of POSIX systems, where only file system objects are protected with a standard set of permissions (read, write, and execute).

E.2 SelectAccess by Baltimore Technologies

SelectAccess is an authorization management solution that provides *Privilege Management Infrastructure (PMI)*, allowing the administration and enforcement of user privileges and transaction entitlements to e-commerce and Enterprise resource. In an extranet environment, SelectAccess provides role-based authorization to Web-based resources, which allows enterprises to provide security and rich user experiences for their customers, suppliers and partners. [Baltimore 2001b]

Privilege Management Infrastructure (PMI) provides the framework to apply policy-based authorization to applications and resources based on a user's business role or relationship to a given organization. Using PMI, an enterprise can be assured that employees, customers, suppliers and partners get the access and transaction authority they need to enable e-business.

One important component of PMI is the Security Policy. The policy defines an organization's top-level direction on information security, including principles for sharing data over the public Internet. The security policy includes statements on how the organization will segment data according to its sensitivity, parties requiring access, and the levels of control (including strength of authentication and encryption) required to match the levels of risk. [Baltimore 2001b]

E.2.1 Component Architecture

Baltimore SelectAccess consists of six core technical components creating a consistent architecture. A secure audit server is also incorporated to track communications between components and/or administrative transactions. By leveraging current programming languages, industry adopted standards, and state-of-the-art security methods, SelectAccess adapts to any existing network infrastructure and can be extended to meet the needs of future security requirements. [Baltimore 2001]

Policy Builder – A graphical user interface used to configure all aspects of SelectAccess. From this central administration point, users and groups are displayed; authorization policies are created; delegated administration is employed; authentication and encryption requirements are defined; and audit and reporting is managed.

Directory Server – Used as a repository for user profiles, resource information and associated policy data. SelectAccess uses any LDAP compliant directory server to interface with existing corporate user data.

Validator – The point at which access requests are accepted and evaluated on behalf of SelectAccess-enabled applications. When a query is received, the Validator accesses the required information from the directory server, interprets the policy logic and conditions are surrounding the request, and returns the decisions to the application. The Validator also determines the administrator's authority over delegated administration.

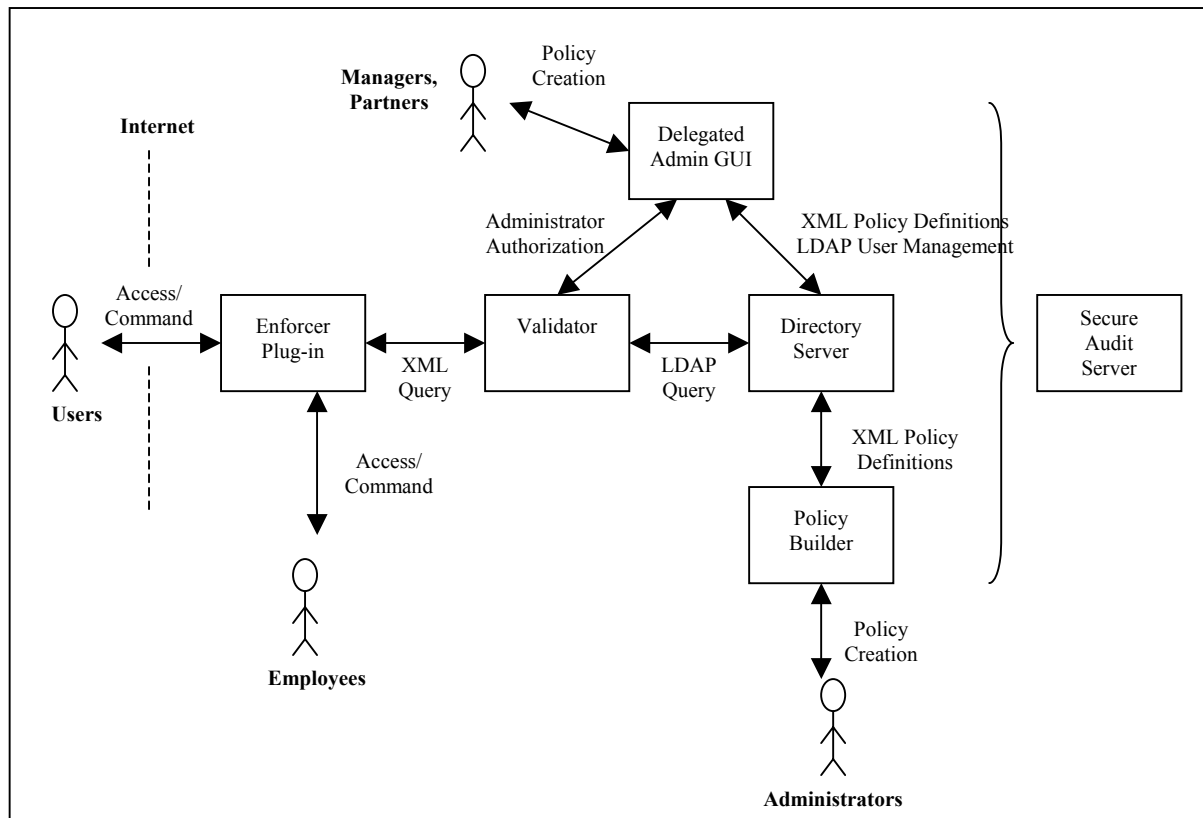


Figure 23: *Component Architecture in SelectAccess.*

Enforcer Plug-In – The plug-in, or agents, installed inside an application. When a user makes an access or transition request, the Enforcer Plug-in queries the Validator for authorization and enforces the returned decision. SelectAccess’s open API framework allows it to be extended in almost every capacity. With API’s, the system can be plugged into any service or application; resource discovery capabilities can be extended; decision criteria on which policy rules are based can be added, as can additional authentication methods.

Delegated Administration GUI – Used to delegate administration of some or all of the user profiles and policy to additional administrators, managed by a web-based interface. Delegated administrators only see those parts of the user and resource space for which they have permission to manage.

Secure Audit Server – A server dedicated to tracking communications between certain components of SelectAccess. All access and authorization requests and responses that are passed to the Validator are logged, and the audit server timestamps and digitally signs the entries. The audit server also reports all policy and administrative changes made to the Policy Builder and Delegated Administration GUI.

E.2.2 Access Control Mechanisms

SelectAccess has the following mechanisms for access control.

Dynamic Role-Based Authorization

SelectAccess provides support for role-based authorization, sometimes referred to as RBAC (Role-Based Access Control). Roles are dynamic groups for which the membership is determined based on changing information. Roles in SelectAccess are defined in the Policy Builder and are displayed in the matrix just as user and group entries would be. Expanding a role triggers the Policy Builder to dynamically determine all the users who are currently a member, and display the membership to the administrator. When the Validator sets its policy cache, it also executes the role queries and dynamically caches the users of that role.

Roles are based on any user information stored as LDAP attributes, and expressed as LDAP queries. The queries can include multiple attributes incorporating Boolean (and/or) logic. For example a “High Spending Customer” role could be defined if a customer has a total monthly spending average or more than \$10 000, and either more than \$10 000, in their pre-paid account or a \$25 000, or greater credit limit. [Baltimore 2001b]

Matrix instead of ACL

In order to scale an interface to the needs of large networks, the ability to easily view, understand and modify policy must be assessed. Most products require users to set policies using ACLs, which execute policy rules in first matching order. In order to understand and update the policy, each rule in the list must be accounted for and understood. Large sites quickly grow of thousands of access rules. Difficulty in determining a resource’s policies can result in configuration errors. These problems hold back policy management and can lead to serious security holes.

SelectAccess provides a visual representation of a company’s access and authorization policy. Users, resources and associated policy rules are displayed in a scalable matrix. This hierarchical matrix not only makes it easy to understand the overall policy, but also makes it easy to see the specific policy for a particular network resource for a given person. Questions like “who can access this URL” are easily answered at a glance.

Unlike access lists, which apply authorization based on the first matching rule in the list, the matrix layout in SelectAccess removes the first matching requirement. The nature of the matrix allows administrators to concentrate on their areas of concern, without having to account for every access rule. The visual nature makes it easy to understand the policy set by other authorized administrators and the management authority delegated to sub-administrators. [Baltimore 2001b]

Authentication and PKI Integration

SelectAccess supports multiple authentication types, including user self-registration, passwords, X.509 certificates, RSA SecurID, two factor tokens, RADIUS, etc. This allows flexibility in the strength of user identification to meet current and future security needs. Passwords are currently the authentication method of choice for extranets, with digital certificates the best solution for scalable authentication. Tight integration with PKI technologies allows authentication of users both inside and outside organizations, without the need to deploy specific hardware-based security solutions. [Baltimore 2001b]

E.3 VPN-1 by Check Point

Check Point's Secure Virtual Network (SVN) architecture meets the Internet security challenges facing companies in the age of E-business. The SVN approach provides secure and seamless Internet connectivity between networks, systems, applications and users across the Internet, intranets and extranets.

The VPN-1 family of products has been developed by Check Point to meet the demanding requirements of enterprise VPNs providing secure Internet-based connectivity to corporate networks, remote and mobile users, satellite offices, and partner sites. Based on Check Point's FireWall-1 security suite and the company's VPN technology, the VPN-1 solutions integrate advanced virtual private networking as a core component of an overall enterprise security policy. [Check Point 2001]

E.3.1 Component Architecture

FireWall-1 – Firewall-1 has a scalable, modular architecture that enables an organization to define and implement a single, centrally managed Security Policy. The enterprise Security Policy is defined at a central management console and downloaded to multiple enforcement points throughout the network.

FireWall-1 consists of the following components:

- FireWall-1 Firewall Module
- Graphical User Interface (GUI)
- Management Server

FireWall-1 Firewall Module – It is deployed on Internet gateways and other network access points. The management server downloads the security policy to the firewall module, which protects the network. Within the firewall module, an inspection module examines every packet passing through key locations in your network (gateways, servers, workstations, routers, or switches), promptly blocking all unwanted communication attempts. Packets do not enter the network unless they comply with the enterprise security policy. [Check Point 2000]

GUI – An enterprise-wide security policy is defined and managed using an intuitive graphical user interface. The security policy is defined in terms of network objects (e.g. hosts, networks, gateways) and security rules. The GUI also includes a log viewer and system status viewer.

Management Server – The security policy is defined using the GUI and saved on the management server. The management server maintains the FireWall-1 databases, including network object definitions, user definitions, the Security Policy, and log files for any number of firewall enforcement points. User information can also be stored in LDAP-enabled directories. The GUI and the management server can be deployed on the same machine or in a client/ server configuration.

VPN-1 Certificate Manager – Integrates best-of-breed technologies into a complete turnkey public key infrastructure (PKI) and user management solution for Check Point IPsec/IKE-compliant VPNs. The Certificate Authority (CA) from Entrust Technologies provides comprehensive key lifecycle management. The LDAP-compliant directory from Netscape Communications stores the X.509 digital certificates for all VPN nodes, as well as the Certification Revocation Lists (CRLs). Check Point Software has pre-configured these industry-leading technologies specifically for VPN-1, and integrated them with a unified installation and management interface. [Check Point 2000]

E.3.2 Access Control Mechanisms

The security server provides authentication for users of FTP, HTTP, TELNET, and RLOGIN. If the security policy specifies authentication for any of these services, the inspection module diverts the connection to the appropriate security server. The security server performs the required authentication. If the authentication is successful, the connection proceeds to the specified destination.

A concept called content security describes access control that is application specific. Content security is available for HTTP, FTP, and SMTP.

- **HTTP** – The HTTP security server provides content security based on schemes (HTTP, FTP, GOPHER, etc.) methods (GET, POST, etc.), hosts (for example “*.com”), paths and queries. A file containing a list of IP addresses and paths to which access will be denied or allowed can be used.
- **FTP** – The FTP security server provides content security based on FTP commands (PUT/GET), file name restrictions, and anti-virus checking for files transferred.
- **SMTP** – The SMTP security server provides content security based on “From” and “To” fields in the mail envelope, header, and attachment types. In addition, it provides a secure e-mail application that prevents direct online connection attacks. The SMTP security server also serves as an SMTP address translator, that is, it can hide real user names from the outside world by rewriting the “From” field, while maintaining connectivity by restoring the correct addresses in the response.

På svenska

Detta dokument hålls tillgängligt på Internet – eller dess framtida ersättare – under en längre tid från publiceringsdatum under förutsättning att inga extraordinära omständigheter uppstår.

Tillgång till dokumentet innebär tillstånd för var och en att läsa, ladda ner, skriva ut enstaka kopior för enskilt bruk och att använda det oförändrat för ickekommersiell forskning och för undervisning. Överföring av upphovsrätten vid en senare tidpunkt kan inte upphäva detta tillstånd. All annan användning av dokumentet kräver upphovsmannens medgivande. För att garantera äktheten, säkerheten och tillgängligheten finns det lösningar av teknisk och administrativ art.

Upphovsmannens ideella rätt innefattar rätt att bli nämnd som upphovsman i den omfattning som god sed kräver vid användning av dokumentet på ovan beskrivna sätt samt skydd mot att dokumentet ändras eller presenteras i sådan form eller i sådant sammanhang som är kränkande för upphovsmannens litterära eller konstnärliga anseende eller egenart.

För ytterligare information om Linköping University Electronic Press se förlagets hemsida <http://www.ep.liu.se/>

In English

The publishers will keep this document online on the Internet - or its possible replacement - for a considerable time from the date of publication barring exceptional circumstances.

The online availability of the document implies a permanent permission for anyone to read, to download, to print out single copies for your own use and to use it unchanged for any non-commercial research and educational purpose. Subsequent transfers of copyright cannot revoke this permission. All other uses of the document are conditional on the consent of the copyright owner. The publisher has taken technical and administrative measures to assure authenticity, security and accessibility.

According to intellectual property law the author has the right to be mentioned when his/her work is accessed as described above and to be protected against infringement.

For additional information about the Linköping University Electronic Press and its procedures for publication and for assurance of document integrity, please refer to its WWW home page: <http://www.ep.liu.se/>

© Erik Boström