THE INTRANET AUTHORIZATION PARADIGM

Mark Vandenwauver
ESAT/COSIC, Kardinaal Mercierlaan 94
3001 Heverlee - BELGIUM
mark.vandenwauver@esat.kuleuven.ac.be

Paul Ashley, Gary Gaskell
Information Security Research Center, School of Data Communications
Queensland University of Technology, GPO Box 2434, Brisbane - AUSTRALIA
ashley@t.qut.edu.au
gaskell@t.qut.edu.au

Abstract: As we approach the new millenium it is clear that the vast increase in the use of information technology will continue well into the next century. Organizations are being reengineered with increasing use of information technology in all aspects of their processes. On the positive side, this helps organizations to become more efficient. The negative side is that new risks are rapidly emerging.

Intranets are the internal computer networks of organizations, and used for their essential business processes. Intranets are at the same time a large asset and a big risk for organizations. Data within these networks can be destroyed, intercepted and even modified during transmission or storage. Such attacks are even more likely when executed by employees or contractors, people inside the organization. This requires a careful design of the security measures to reduce the risk to the organization.

This paper compares two solutions that provide advanced security functionality toIntranets. The first is the DCE technology from the Open Group. Although this technology has been available and deployed for some time it is still evolving to satisfy the requirements of modern Intranets. The second is SESAME, a relatively new technology that is very rich in security services. Both of these technologies provide a big advantage over other technologies: a centralized management system for authorization. It is the centralised administration of security privileges that sets these security architectures apart from other secure technologies such as SSH or Kerberos. This paper will explain how this is achieved and why this is so significant in the Intranet situation.

Key words: Access control, authorization, DCE, Intranet, SESAME.

The original version of this chapter was revised: The copyright line was incorrect. This has been corrected. The Erratum to this chapter is available at DOI: 10.1007/978-0-387-35575-7_19

J. H. P. Eloff et al. (eds.), *Information Security Management & Small Systems Security*

1. INTRODUCTION

An Intranet is an organization's internal computer network, frequently built using Internet technologies. Most of the time Intranets are either connected directly to the Internet or the Internet is used to connect various Intranets into one common global Intranet. To date much of the focus in security research has been on the Internet. This is mainly due to the potential for its use in electronic commerce. The security of the Intranet however is equally important for the survival of organizations. The Intranet is where the essential processes of an organization occur, and they must be protected from attacks.

Securing an Intranet can be very difficult. Employees and contractors are inside the organization and have much better access to resources than people outside the organization. Hence protecting the Intranet from internal attack requires a very careful design. There is also the concern that security management must be as efficient as possible. Solutions that are unwieldly and costly to manage are often rejected as being not worth the effort.

Since the late 80's various researchers have looked for security architectures that could be implemented across internal networks. Their work can now be adapted to Intranets. The best known and most used solution until now has been the Kerberos [NT94] scheme introduced at MIT. Several other suggestions have been made such as KryptoKnight [JTY97], Yaksha [Gan95] and DASS/SPX [TA91]. All of these solutions have the same limitation. They provide excellent user and network entity authentication but management of authorization is difficult and cannot be done in a cost effective way. In addition SSL/TLS [DA99] has been developed. It provides network entity authentication and was designed to secure the channel. Authentication remains important as a good and reliable means of authentication constitutes a necessary first step to provide authorization. However, the previous schemes rely on *identity* based authorization models, and this type of scheme does not scale well to large organizations.

DCE and SESAME both provide a (partially) centrally manageable authorization scheme and this is described and compared in this paper. The benefit of centrally manageable authorization cannot be underestimated, as it allows the technologies to scale to very large Intranets, something that is becoming more common everyday. SESAME and DCE also provide additional functionality, for example providing an authorization component to verify authorization information and provide some level of access control at the resource.

The paper is organized as follows. It begins by giving a high level description of both DCE and SESAME. They are then compared to show how they differ in the provision of authorization. The paper finishes with our conclusions.

2. DCE

The Distributed Computing Environment (DCE) [Har92, RKF92] was developed by the Open Software Foundation, a non-profit consortium promoting the development of open computing. Development of DCE began in the late 80s, and there has been much renewed interest lately. The goal of DCE is to provide a vendor-neutral open middleware, with suitable security. Lately OSF has joined forces with the X/Open group to form a new organization called The Open Group (TOG). TOG looks after the ongoing development of DCE. The latest public release is version 1.2.2. The system is also freely publicly available though the licensing permits only internal use. The free release of DCE is hoped by many to help DCE regain the ante in the battle for the defacto middleware standard that has been lost to the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA).

The goal of DCE security is to provide a security infrastructure for the DCE architecture. In particular, DCE security has four main goals:

1. To allow a user a single sign-on to the network (user authentication);
2. To provide authentication of network entities;
3. To provide data confidentiality and data authentication;
4. To provide an authorization service;

DCE's security is based on Kerberos V5 which provides all of the above services except for an authorization service. Authorization in Kerberos V5 is based on the user's identity which is carried in the service ticket and implemented when the application server checks an Access Control List (ACL). As Kerberos only authenticates users, a separate entry for every user must be made in the ACL. In this way Kerberos does not scale well.

The authorization in DCE is only in part centrally manageable. In fact it is exactly like the Unix file system with access control lists (ACLs). The groups that people are in is centrally managed, but the ACLs are managed by the administrators of the particular application. The difference between Kerberos and DCE is that the ACL in DCE can include groups and users. As most users can be granted access according to their

group memberships, the ACL needs to contain only a list of groups and a few *special* users. Importantly though, as the access privileges rarely change (when compared to staff movement) the ACL rarely has to be updated and the vast majority of authorization administration can be performed centrally.

DCE provides authorization support. Initially it provides the secure distribution of group membership. It provides this to the application, via a library service that does all the decryption and ticket verification. The initial release of DCE did not include library calls for the management of ACLs, as it was believed that using the POSIX ACL definition ACL management, code/tools were already available. These library calls were added however in the more recent releases.

A problem with DCE authorization is that it does not provide non-repudiation as it uses symmetric key technology.

2.1 PROTOCOLS

DCE security is based on Kerberos V5 with an authorization service added [KPS95]. The protocol is designed to be interoperable with Kerberos V5.

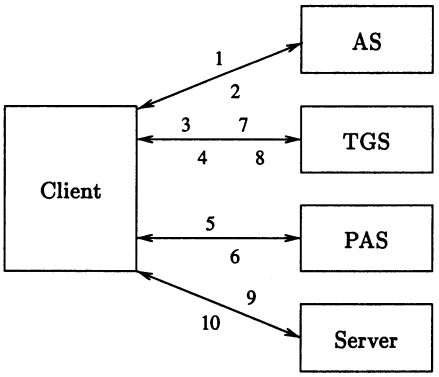

Figure 1 Overview of DCE Security

Figure 1 illustrates the architecture of DCE:

1. A user logs into the Authentication Server (AS);
2. The AS returns a Ticket Granting Ticket (TGT) to the Client;
3. The user presents the TGT to the Ticket Granting Server (TGS) and requests a ticket to the PAS;
4. The TGS returns a ticket to the Privilege Attribute Server (PAS) called the Privilege TGT (PTGT). To keep DCE compatible with

Kerberos V5, the TGS views the PAS as any other application server;

5. The Client presents the PTGT to the PAS;
6. The PAS returns the PTGT to the Client with the user's identifier (UID) along with the groups (GIDs) the user is a member of. This is the completion of the login process;
7. When a Client wants to access an application server (Server), the Client supplies the TGS with the PTGT and requests a ticket to the Application Server;
8. The TGS returns a Server ticket to the Client;
9. The Client presents the Server ticket to the Server containing the user's UID and GID information;
10. The server's verification module checks the ticket and extracts the User's information (UID, GID). It then matches this information with its local ACLs to check whether access should be granted. Similarly to Kerberos V5, the server can also be mutually authenticated to the Client.

The whole process is similar to Kerberos V5 except that Server tickets contain authorization information and there is an additional access to the PAS. A session key is established similarly to Kerberos V5 and this is used by DCE's Remote Procedure Call (RPC) mechanism for secure client/server communication. Also similarly to Kerberos, it is possible to require the use of *sub-session keys*. This is useful to prevent *dictionary attacks* if the application data is highly formatted.

Not shown on the figure is another component of DCE called the registration server. The purpose of the registration server is to set up the registration database used by the AS, TGS and PAS.

More recent DCE has included a facility known as *PKINIT* [SH98] which is targeted to incorporate the partial use of public-key technology within DCE. As this is not used in the authorisation system of DCE, it is not considered further in this paper.

3. SESAME

SESAME [AV99] is the name of a security architecture. It is the result of a collaboration of Bull, ICL and Siemens together with some leading European research groups [Kai98]. SESAME is an acronym for *"A Secure European System for Applications in a Multi-vendor Environment"*.

Figure 2 gives an overview of the SESAME architecture. At first glance it looks very complex but it is possible to distinguish three boundaries in the architecture: the client, the domain security server, and the (application) server.

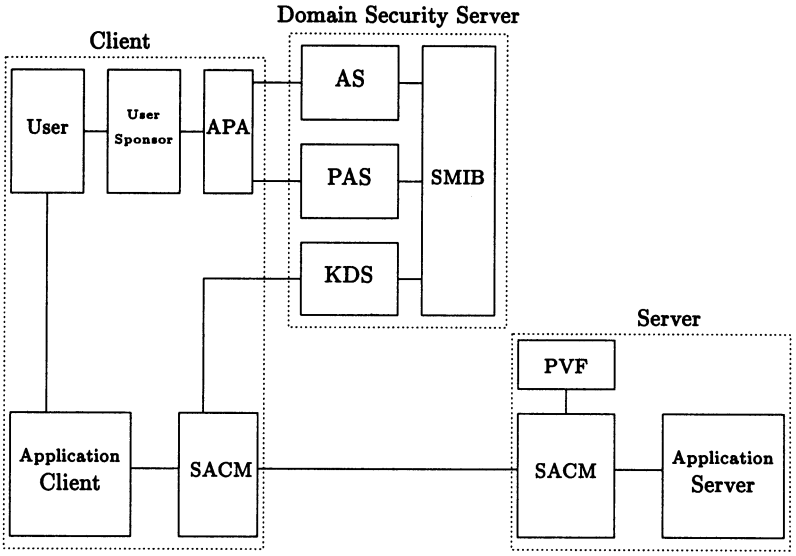

Figure 2 Overview of the SESAME components

The client system incorporates the User, User Sponsor (US), Authentication Privilege Attribute (APA) Client, Secure Association Context Manager (SACM) and client application code. The User Sponsor gives the user the interface to the SESAME system, and allows the user to log on. The APA is used by the User Sponsor for the communication with the domain security server. The SACM provides the data protection services (data authentication, data confidentiality, non-repudiation) for the client-server interaction.

The Domain Security Server is very similar to Kerberos V5 [NT94]. The main difference is the presence of the Privilege Attribute Server (PAS) in SESAME. The server has been added to manage the access control mechanism that is implemented by SESAME. SESAME has opted to implement role based access control (RBAC) [SCFY96]. This scheme is enforced using Privilege Attribute Certificates (PACs) [ECM96]. The function of the Authentication Server (AS) and Key Distribution Server (KDS) (TGS in Kerberos V5) are similar to their Kerberos V5 counterparts: providing a single sign-on and managing the cryptographic keys. A major difference with Kerberos V5 is that SESAME also supports public-key based authentication using the X.509 authentication protocol [ITU93].

When the application server receives a message from an application client, indicating that it wants to set up a secure connection, it forwards the client's credentials and keying material to the PAC Validation Facility (PVF). The PVF checks whether the client has access to the application. If this check is successful, it decrypts the keying material and forwards the session keys to the SACM on the server machine. Both PVF and SACM run on the same machine as the application server and it is supposed that the communication between them is secured by the operating system.

SESAME uses independent keys for providing data authentication and data confidentiality and can thus easily be adapted to local legislation. This is an important feature of SESAME that distinguishes it from other architectures. Using the SACM on the application server, the application authenticates to the client (mutual authentication). This SACM also enables the application server to secure its communication with the application client.

In contrast to the DCE implementation, SESAME has opted to implement the PAC Validation Facility as a separate daemon (in DCE it is a library call). Taking this functionality out of the application process renders it more easy to check this security critical code and verify whether it works properly and makes it safer from tampering.

A detailed description of SESAME including the protocols and applications secured with SESAME can be found in [AV99].

4. AUTHORIZATION

In general securing a computer networks consists of two stages:

1. Determine who the user or entity is that is trying to access a resource on the network.

2. Decide whether this person or entity is allowed to access this specific resource.

The first step is defined as *user or entity authentication.* While entity authentication is the essential first step in providing security in an organization, it is certainly not the only security service that is required.

The second step, deciding what an authenticated entity is allowed to do, is what we define as *authorization.* One common way of approaching this problem is by implementing access control to each resource connected to the network. In the actual business model it is indeed necessary to be able to decide what entities are allowed to do and thus implement authorization.

Both DCE and SESAME have chosen to solve this problem by providing separate services. In the next sections it is described how DCE and SESAME have implemented the authorization service. Although they use a similar approach there are also some considerable differences.

4.1 DCE

DCE's approach to authorization was shown in Figure 1. Part of the user login process involves having the authorization field of the TGT filled in by the PAS. This field is called the DCE PAC. The security attributes in the DCE PAC contain the entity's identity and group-related information. The DCE PAC is shown in Table 1.

Table 1 DCE Privilege Attribute Certificate

Authentication Flag
Cell UUID
Principal UUID
Primary Group UUID
Secondary Group UUIDs
Foreign Group UUIDs

The *authentication flag* identifies whether the certificate was authenticated by the DCE Authentication Server (a client is able to present unauthenticated PACs). The *Cell UUID* is the cell in which the entity is registered. DCE uses the term *cell* to describe a Kerberos *realm* and the term Universal Unique Identifier (UUID) to uniquely identify every resource in the system. The principal UUID identifies the entity whose privileges are stored in the PAC. The rest of the PAC contains the *groups* that the entity is a member of.

DCE provides a server-side component for verifying the PAC using authorization libraries. These are built in to the application using the provided Application Programming Interface (API). This is one of the differences with SESAME.

The PAC (see Table 1) is protected from theft because it is part of the Kerberos V5 ticket and in the usual Kerberos V5 way this is protected with only the client and server having access to the symmetric key in which it is encrypted.

From DCE version 1.1 delegation of PACs has been supported. This allows a client to delegate its PAC to a server, and the server then to act as a client to another server. This delegation unfortunately is unconditional in contrast with the delegation feature provided by SESAME.

Table 2 The SESAME Privilege Attribute Certificate

Common Contents	Specific Contents	Signature
issuer domain issuer identity serial number creation time validity	protection methods privileges miscellaneous time period	value of the signature algorithm identifier certificate information

Many organizations use the groups facility within DCE to implement an RBAC mechanism. The RBAC deployment is dependant upon the administrators assigning users to the groups according to the roles that they perform in an organization. It will be described in a later section how the RBAC system in SESAME is richer.

4.2 SESAME

To address the authorization problem SESAME opted for RBAC. RBAC has many advantages, according to Ferrailolo and Kuhn [FK92], and Sandhu et al.[SCFY96]:

- It provides a means of administering the access to data that is natural and consistent with the way most enterprises are organized and conduct business;
- the users can be granted membership to roles based on their competence and responsibility;
- user memberships can be revoked or re-established at any time;
- the administrative functions can be applied centrally, locally or remotely;
- administrators can create roles and place constraints on role memberships;
- it is possible to define role hierarchies.

RBAC is implemented by SESAME using attribute certificates. Attribute certificates are tokens that contain the role(s) that users are allowed to act as. In this way they give the appropriate privileges to the user. This is why these attribute certificates have been defined as PACs in SESAME (the major fields of the PAC are illustrated in Table 2). They are issued and managed by a central server (the PAS, see

Figure 2). To prevent these PACs from being forged by the user or by outside attackers, they are digitally signed with the PAS's private key.

The access control decision in SESAME is implemented on the application server. This server checks an ACL for an entry for role or user identity.

At the initiator end, privilege attributes and the controls over the PAC's use are chosen on the basis of the user's authenticated identity and the role name. This may result in any valid combination of attributes. In addition, both for pre-defined attributes and for administrator defined ones, the application may request specific PACs that only contain a subset of the permitted attributes (principle of least privilege [KPP94]).

It is possible to distinguish between two types of PACs:

1. *Non-delegatable*: these are bound to a particular identity. They are protected by the Primary Principal Identifier (PPID) method (described later). The security information inside the PAC gives more information about the particular session.

2. *Delegatable*: these act like capabilities. It is thus possible to temporarily delegate some of a user's access rights to another server. It remains important to keep this server accountable for its actions. Therefore, each entity in the system has its own identity, and is always authenticated as that identity. To implement this delegation mechanism, the PV/CV mechanism (described later) is used. In general it is good practice to make the rights that are conveyed as restricted as possible.

Both types of PAC are issued with short expiration times (the order of a few hours) to limit the time a compromised key or capability can be used for. When an access control decision is presented with a PAC, the target (more precisely its PVF) checks that the PAC is currently valid. The time period during which a PAC is valid is intended to be short so that the User Sponsor must periodically re-contact the PAS. This ensures that changes to a user's privilege attributes are guaranteed to take effect within a known, and short, period of time thus fulfilling the need for a possible revocation of the user's rights.

The PAC format is independent of the domain's security policy. The details of the security policy are contained in the system components that create or interpret PACs: the PAS and each application server's and PVF's access control logic. The SESAME implementation assumes a particular form of role-based policy: for (and during) a particular session, each user takes on exactly one role; roles are enumerated and assigned identifiers; for each user, there is a list of the roles in which the

user can act; the access rights of a user are determined by the role in which they act.

The SESAME PAC is based on a profile of the ECMA PAC. A complete definition of the ECMA PAC can be found in ECMA-219 [ECM96].

4.2.1 Primary Principal Identifier (PPID). In order to prevent a PAC from being used in an unauthorized manner, the concept of PAC ownership has been introduced. The protection method is known as the *PPID Protection method.* This method allows the PAC to be used securely from the original owner's workstation at more than one target, even though the targets concerned may not be trusted not to attempt to use the PAC as if they were its owner. Unless delegation is separately permitted (using the PV/CV method described in Section 4.2.2) none of the potential receiving targets can pretend to be its owner or act as delegate for its owner.

The PPID method controls the use of a PAC by putting an identifier (the PPID) for the primary principal initiating the request for the PAC in the PAC itself, and supplying the same information as part of the key establishment information. This enables a target application server to ensure that the entity sending the PAC is the same entity as the one that obtained the keying information. This achieves the necessary protection and even if it is possible for a wire-tapper to intercept the PAC, any intercepted keying information cannot be sensibly used or forged.

4.2.2 Protection Value/Check Value. The PV/CV protection method allows a PAC to be used by proxy: passed from the initiator to a delegate, and then from delegate to delegate or final target. Each delegate then has the capability of issuing new actions to the applications for which the PAC is valid.

In this method, valid initiators are linked to the PAC by means of a *Protection Value* (PV), inserted in the PAC. The PV has a corresponding randomly generated *Check Value* (CV). The protection value is the result of a one-way function applied to the check value. The only initiator that initially knows the CV is the original requester of the PAC.

$$PV = f(CV).$$

In SESAME, PV/CV pairs are generated by the PAS. The CV is returned to the initiator encrypted under the appropriate session key. The initiator, and subsequently its valid delegates, prove knowledge of the CV for a particular PV by passing it encrypted under the current session key. Each receiving target can then use the PAC by proxy (subject to some limitations) since its PVF has now learned the corresponding CV.

Delegation can therefore be permitted and controlled without the original initiator needing to be aware of the precise identity of the final target application server or the route to it.

4.2.3 (Delegate) Target Qualifier (DTQ). A SESAME PAC may contain one or more target, delegate-target application and/or *Trust Group* names specifying which targets or delegate-targets the PAC is valid for. A trust group name is simply the name of a group of applications, defined by the security administrator, that mutually trust one another not to spoof one another's identities. The control information is specified in a *Target Qualifier* or *Delegate/Target Qualifier* protection method (referred to as the DTQ method) which may be used together with either the PPID or PV/CV protection method.

The presence of the DTQ method in the same group as the PPID or PV/CV method serves to limit the acceptability of the PPID check or PV/CV check to be only acceptable for the targets or delegate-targets identified by the DTQ method. If no DTQ method field is present, the PAC is acceptable by any PVF provided that it passes the other controls. If a DTQ method is present, a PVF checks this field against the identity of the application, or its Trust Group, on whose behalf the target is making the PAC validation request. The privilege attributes contained in the PAC are valid if one of the protection methods (PPID or PV/CV) is accepted and if the DTQ method also passes. There are two ways to pass the DTQ method controls: there is no target qualifier, or the identity of the application or its trust group matches a DTQ method value.

In SESAME, there is a possible distinction between targets that are permitted also to be delegates, and targets that are not. For that purpose the DTQ method field may specify that the identities contained in it are the identities of targets only or of delegate-targets. An application that is only nominated as a target is not permitted to act as a delegate. Thus to be able to act as a delegate, an application must be nominated as a delegate-target. If a target is not permitted to be a delegate as well, the PVF does not return any received check value.

5. COMPARISON OF THE AUTHORIZATION MODELS

Role Base Access Control (RBAC) is a very important technique to manage an Intranet's authorization system. RBAC is very well suited to an application with a large number of users with overlapping requirements [NO93]. As Intranets commonly have large numbers of users,

how DCE and SESAME implement RBAC will be a main focus of this analysis.

Table 3 summarizes the differences in the authorization models of DCE and SESAME. Both systems implement a PAC that is retrieved from a PAS to be presented to a remote resource. In both cases the PAC contains authorization information that aids the scaling of systems with large numbers of users, whereas Kerberos only authenticates the user's identity. In very large organizations it becomes impractical to base access decisions solely on identity, as Kerberos supports. DCE implements its privileges with groups, whereas SESAME uses roles. While it is possible to implement RBAC with groups in DCE, the management facilities and options are not as rich as in SESAME.

A system design for role inheritance within DCE was presented by IBM Germany to the DCE security community in 1994 [LG94]. The system design used the approach of controlling group membership to implement RBAC, role inheritance, the abstraction of roles into "role types" and a concept known as "resource sets". This was achieved using the DCE core functionality of groups and enforced by controlling all group memberships via the "role administration tool". However many installations resort to manual systems to implement RBAC via DCE's groups.

DCE's PAC is protected through inclusion within the Kerberos-based ticket. This uses symmetric key technologies. SESAME protects it's PAC by the use of a digital signature (public-key technology). DCE and SESAME provide delegation of their PACs, although delegation in DCE is uncontrolled. DCE and SESAME also both verify their PACs.

Table 3 DCE and SESAME Authorization Models

Implementation	DCE	SESAME
PAC	x	x
Groups	x	x
Roles	(x)	x
Symmetric Protected PAC	x	
Asymmetric Protected PAC		x
Delegation of PAC	x	x
Trusted Entity at resource for Verifying PAC	x	x
Trusted Entity at resource for Controlled Delegation of PAC		x

A limitation of both DCE and SESAME is the dynamic acquisition or disposal of a role (group in the case of DCE). Dynamic acquisition of

roles [NO93] is particularly important in Object Oriented (OO) systems where the number of subjects and objects can be very large.

This is an issue that should be investigated in the future, as it appears to be a common request from industry, particularly the finance industry. It is appears to be a common requirement to be able to instantly revoke an individual's access.

6. CONCLUSION

DCE and SESAME both provide authorization models suited to a large Intranet environment. The SESAME model may be considered superior to the current DCE model, in its use of public-key cryptography and the existence of a trusted entity at the resource to assist delegation. However this use of different cryptographic primitives does not concern most systems integrators. It is important though in environments when higher assurance of the system security is required. It can be seen that the authorization services provided by DCE are very similar to SESAME. This is not surprising considering the proposals submitted to the Open Software Foundation by the early developers of SESAME [Fai92]. Furthermore, recent initiatives in the Open Group indicate that the next version of DCE will migrate even closer to the SESAME authorization model.

References

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

[DA99] T. Dierks and C. Allen. The TLS Protocol Version 1.0, January 1999. RFC2246.

[ECM96] ECMA 219. ECMA-219 Security in Open Systems - Authentication and Privilege Attribute Security Application with Related Key Distribution Functionality, 2nd Edition, March 1996. European Computer Manufacturers Association.

[Fai92] B. Fairthorne. Security Enhancements for DCE 1.1. Technical Report OSF-RFC 19.0, ICL/SESAME and Open Software Foundation, 1992.

[FK92] D.F. Ferraiolo and R. Kuhn. Role-Based Access Control. In *Proceedings of the 15th NIST-NSA National Computer Security Conference*, Baltimore, MD., October 1992.

[Gan95] R. Ganesan. Yaksha : Augmenting Kerberos With Public Key Cryptography. In *Proceedings of the Internet Society Sym-*

posium on Network and Distributed System Security, pages 132–143, February 1995.

[Har92] D. Hartman. Unclogging Distributed Computing. *IEEE Spectrum*, 29(5):36–39, May 1992.

[ITU93] ITU. ITU-T Rec. X.509 (revised). The Directory - Authentication Framework, 1993. International Telecommunication Union, Geneva, Switzerland.

[JTY97] P. Janson, G. Tsudik, and M. Yung. Scalability and Flexibility in Authentication Services: The KryptoKnight Approach. In *Proceedings of IEEE Infocom '97*, 1997.

[Kai98] P. Kaijser. A review of the SESAME Development. In C. Boyd and E. Dawson, editors, *Proceedings of the 3rd ACISP Conference - LNCS 1438*, pages 1–8. Springer-Verlag, 1998.

[KPP94] P. Kaijser, T. Parker, and D. Pinkas. SESAME: The Solution To Security for Open Distributed Systems. *Computer Communications*, 17(7):501–518, July 1994.

[KPS95] C. Kaufman, R. Perlman, and M. Speciner. *Network Security: Private Communication in a Public World.* Prentice Hall, Inc., 1995. DCE Security Review, pages 455–459.

[LG94] S. Lorenz and V. Gligor. Role-Base Authorization in DCE, July 1994.

[NO93] M. Nyanchama and S. Osborn. Role-Based Security: Pros, Cons & Some Research Directions. *ACM SIGSAC Review*, pages 11–17, 1993.

[NT94] B. Neuman and T. Ts'o. Kerberos : An Authentication Service for Computer Networks. *IEEE Communication Magazine*, 32(9):33–38, September 1994.

[RKF92] W. Rosenberry, D. Kenney, and G. Fisher. *Understanding DCE.* O'Reilly & Associates, Inc., 1992.

[SCFY96] R. Sandhu, E.J. Coyne, H.L. Feinstein, and C.E. Youman. Role-Based Access Control Models. *IEEE Computer*, pages 38–47, February 1996.

[SH98] F. Siebenlist and D. Hemsath. Public Key Certificate Login - Functional Specification, July 1998. TOG-RFC 68.4.

[TA91] J. Tardo and K. Alagappan. SPX: Global Authentication Using Public Key Certificates. In *Proceedings of the 1991 IEEE Symposium on Security and Privacy*, pages 232–244, May 1991.