ECMA

EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

SECURITY IN OPEN SYSTEMS
A SECURITY FRAMEWORK

ECMA TR/46

July 1988

Free copies of this document are available from ECMA,
European Computer Manufacturers Association
114 Rue du Rhône – 1204 Geneva (Switzerland)

ECMA
EUROPEAN COMPUTER MANUFACTURERS ASSOCIATION

SECURITY IN OPEN SYSTEMS
A SECURITY FRAMEWORK

ECMA TR/46

July 1988

Brief History

ECMA, ISO and CCITT are working on standards for distributed applications in an Open System environment. Examples are the OSI Reference Model, the work on Open Distributed Processing and the Framework for Distributed Office Applications.

Security is a major concern in information processing. The security aspects of interconnection have been addressed by ISO in the work on the OSI Reference Model (DIS 7498/2, Security Architecture). The purpose of this Technical Report is to provide a Framework for the development of security provisions in the Application Layer. This Framework unifies many views of security needs and of security functionality including notions about end-systems security and therefore it allows a coherent approach to the specification of protocols and protocol elements as needed to realize secure Open Systems.

This Report gives an overview of security needs and of the basic functionality needed to answer these needs. Using a generic building block approach it shows how supportive security applications may be constructed to satisfy a wide range of uses. In doing so this Report makes extensive use of the concepts developed in ECMA TR/42, Framework for Distributed Office Applications as well as in ISO/OSI standards. However, other concepts such as the Object Model of processing used in the work of ECMA/TC32-TG2 on the Distributed Application Services Environment, may also be used to describe the security functions developed in this document.

This Report is one of a set of Standards and Reports for Open Systems Interconnection. Open Systems Interconnection standards are intended to facilitate homogeneous interconnection between heterogeneous information processing systems. This Report is within the framework for the coordination of standards for Open Systems Interconnection which is defined by ISO 7498.

This Report is based on the practical experience of ECMA member Companies worldwide and on the results of their active participation in the work of ISO and CCITT as well as in national standards bodies in Europe and the USA.It represents a pragmatic, widely based consensus.

This Report emphasises the need for specification of the externally visible and verifiable characteristics needed for the communication of security related information. However, it avoids placing unnecessary constraints upon the internal design and implementation of information processing system that process and exchange security related information.

This Report is oriented towards urgent and well understood needs and supports rapid and effective standardization. It is intended to be capable of extensions to cover future developments in technology and needs.

Adopted as an ECMA Technical Report by the General Assembly of June 30, 1988.

TABLE OF CONTENTS

1. INTRODUCTION

In recent years, advances in computing and telecommunications technology have greatly expanded the tools available to all users of data processing systems, irrespective of the field of application. This development is paralleled by the emergence of facilities for the distributed processing of application tasks, thus giving users great flexibility in the structuring of their systems and in the interaction with other systems. As a consequence, user organizations are becoming more and more dependent on the services provided by their systems. Increasingly, information of high value, possibly critical to the survival of the organization, is placed on computer systems and exchanged over telecommunications facilities. This trend raises the need for dependable systems that process information securely.

This Report defines a Framework for the development of standards that support a wide variety of security requirements in a multi-user, multi-vendor systems environment. Major objectives in the development of such standards are:

- to allow effective interworking of diverse products
- to allow modular, expandable development of products
- to facilitate implementation.

This report is structured as follows:

- Clause 1 (this Clause) gives a general introduction, references and definitions of terms,
- Clause 2 gives an overview of security requirements from both the operational and from the functional point of view. It also gives implementation considerations and design requirements relevant to the design of secure systems on the basis of this Framework,
- Clauses 3, 4, 5 and 6 describe the Security Framework: the Security Domain concept, the Security Facilities concepts, and the mapping of these concepts to other architectures such as the OSI Reference Model and the Distributed Office Applications Framework,
- Clause 7 describes the management aspects of the security functions introduced in the preceding Clauses.
- Clause 8 gives a summary and conclusions.

1.1 Need and Application

Applications may be distributed for various reasons such as sharing of costly resources (e.g a printer) or distributing functionality (e.g. electronic mail services). Standards for Open Systems Interconnection permit the functional components of applications to be distributed over a network. This must be done in a secure fashion that assures that users can depend on the services provided and the information stored and processed.

Generally, security refers to a complex of measures of procedural, logical and physical measures aimed at prevention, detection and correction of certain kinds of misuse e.g. together with the tools to install, operate and maintain these measures. For the purpose of this report "security" will refer to characteristics of data processing systems that give resistance to attack and misuse, intentional or otherwise. Other aspects of systems security such as reliability, availability and redundancy, are outside the scope of this report.

Given the above definition, security addresses not only attacks and threats originating externally, i.e. by persons not belonging to the organization operating a given network or system, it also addresses internal attacks and threats coming from known persons. By providing guarantees of integrity and or confidentiality of information, secure systems may be used to perform business transactions in such a manner as not to expose their users to unacceptable liabilities. Already, major insurance companies are using higher rates for customers with insecure computer systems.

Secure systems may more easily survive system failures because the tools and mechanisms needed to assure the integrity of information are available.

More and more computers are linked together in systems that provide a wide variety of services to their users. Such systems are frequently referred to as distributed processing systems because single task may require cooperation between processes executing on several end-systems. This Report provides unifying principles, structuring distributed security functions and the associated protocols. This allows a secure environment to be created in which other types of applications may be executed.

1.2 Scope of Security in this Report

Many different security needs can be met by a common set of secure functions to be provided outside application processes. These functions will affect the interactions between users and productive applications, and between productive applications and supportive applications. They will also affect the installation, maintenance and management of applications and of the underlying system. These functions, their interactions and their management constitute the scope of security in this Report.

The level of view addressed in this Report is the level of the "secure environment".This has close parallels with the concept of Open Distributed Processing. The security requirements of distributed applications that are specific to the nature of these applications (e.g. access controls to the objects owned by a given application) are addressed here only to the extent that generally applicable functions and their interactions can be identified.

Where appropriate, this Framework refers to Security Services defined by the OSI Reference Model as defined in ISO 7498/2.

1.3 The Application Layer Security Framework

This document describes a Security Framework in terms of Application Layer functions necessary to build secure Open Systems. Figure 1 illustrates the concept of a secure, distributed system.

To the users and owners, the value represented by computer systems lies mostly in the information residing on these systems and in the application software processing this information. The information will exist in various forms including files on magnetic media and messages transmitted by electronic means. In the figure, this information - the application data - is indicated as "Security Objects".

A secure system protects the application data it processes as well as the application software that performs the processing. It protects information from misuse by users and from misuse by application software. In the figure, users and active applications are indicated as "Security Subjects". (A passive application is a Security Object).

In the Security Framework, the access of Security Subjects to Security Objects is mediated and controlled by Security Facilities. This concept can be applied equally well to the access of users to application software as to access by applications to their data.

The task of the Security Facilities is twofold:

- to protect the integrity and confidentiality of Security Objects (data and programs) in the logical sense, e.g. by limiting access to certain users and applications,
- to protect the integrity and confidentiality of Security Objects (data and programs) where they are subject to physical or other external attacks e.g. when stored on magnetic media or when transmitted over communications links.

The diagram emphasizes that:

- access of Subject to an Object is always via the Security Facilities, (Arrow A),

- access of a Subject to a remote Object using the communications services of a distributed system, is always via the Security Facilities (Arrow B).

"Security" is not a monolith, it can be decomposed into functional elements here referred to as Security Facilities. Clause 3 describes a complete set of ten Security Facilities: the Subject Sponsor, the Authentication Facility, the Association Management Facility, The Security State Facility, the Security Attribute Facility, the Authorization Facility, the Interdomain Facility, the Security Audit Facility, the Security Recovery Facility and the Cryptographic Support Facility.

Implementations of Secure Open systems may implement a variety of security related services. Any security service can be build using one or more of these Facilities.

Not visible in the Figure is the management aspect. All functions require management. In this document "management" is treated as an integral aspect of each of the security facilities identified.

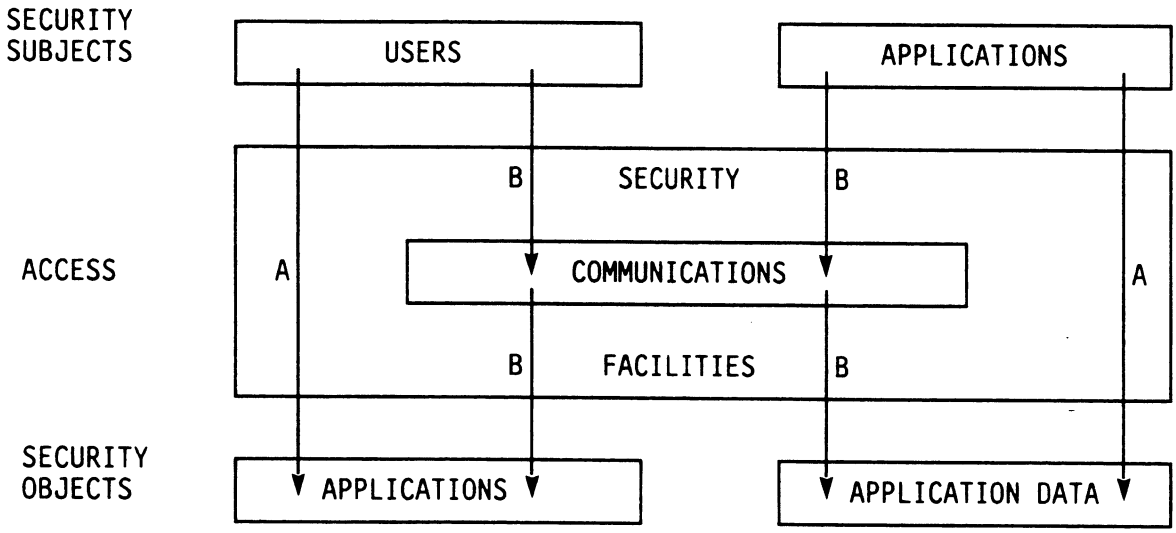


Figure 1 - Conceptual View of the Security Framework

1.4 References

ECMA/TR 31	Remote Operations, Concepts, Notation and Connection-oriented mappings
ECMA/TR 32	Directory Access and Service Protocol
ECMA/TR 37	Management Framework for Open Systems Interconnection
ECMA/TR 42	Framework for Distributed Office Applications
ECMA-93	Distributed Application for Message Interchange (MIDA)
ECMA-122	Mailbox Service Description
ISO 7498	Open Systems Interconnection, Basic Reference Model
ISO 7498/2	Basic Reference Model, Security Architecture
ISO 8649/2	Open Systems Interconnection, Application Service Elements, Part 2 Association Control
ISO 8730	Banking - Requirements for Message Authentication
ISO 8732	Wholesale Financial Institution Key Management
ISO 8824	Specification of Abstract Syntax Notation One (ASN.1)

ISO 8825	Specification of Basic Encoding Rules for ASN.1
ISO 9594/1-8	The Directory
ISO 9545.1	Open Systems Interconnection, Application Layer Structure
ISO 9302/2	Remote Operations: Model Notation and Service Definition
CCITT Rec. X.400	Message Handling System (Rec. X.400 to Rec. X.420)

1.5 Definitions

1.5.1 General Terminology

The following terms are used with meanings defined in ISO 7498:

Application Layer
Application Process
Application Entity
Application Service Element
Presentation Layer
Presentation Connection
Protocol
Service Definition

The following terms are used with meanings defined in ISO 8824.2:

Macro
Macro notation
Coordinated Universal Time

The following terms are used with meanings defined in ECMA TR/31:

Association Control Service Element
Binding
Object
(Note: the text uses "object" or "security object" with lower case "o" with a different meaning.)
Object Instance
Object Type
Operation
Remote Operation
Remote Operation Service Element

1.5.2 Specific Terminology

For the purpose of this Technical Report the following definitions apply.

1.5.2.1 access control

The prevention of use of a resource by unidentified and/or unauthorized entities in any other than an authorized manner.

1.5.2.2 access control list

A set of control attributes. It is a list, associated with a security object or a group of security objects. The list contains the names of security subjects and the type of access that may be granted.

1.5.2.3 access control policy

A set of rules, part of a security policy, by which human users, or their representatives, are authenticated and by which access by these users to applications and other services and security objects is granted or denied.

1.5.2.4 access context

The context, in terms of such variables as location, time of day, level of security of the underlying associations, etc, in which an access to a security object is made.

1.5.2.5 authentication policy

A set of rules, part of an access control policy, by which credentials are matched against claims of identity.

1.5.2.6 authorization

The granting of access to a security object.

1.5.2.7 authorization policy

A set of rules, part of an access control policy, by which access by security subjects to security objects is granted or denied. An authorization policy may be defined in terms of access control lists, capabilities or attributes assigned to security subjects, security objects or both.

1.5.2.8 capability

As used in the context of security, a form of privilege attribute. It is a token (recognized by a process managing access to security objects) possession of which grants access to the security objects to which the token applies.

1.5.2.9 confidentiality

A security property of an object that prevents:

- its existence being known and/or

- its content being known.

This property is relative to some subject population and to some agreed degree of security.

1.5.2.10 control attributes

Attributes, associated with a security object that, when matched against the privilege attributes of a security subject, are used to grant or deny access to the security object.

1.5.2.11 credentials

Data that serve to establish the claimed identity of a security subject relative to a given security domain.

1.5.2.12 distributed application

A set of information processing resources distributed over one or more open systems which provides a well defined set of functionality to (human) users, to assist a given (office) task.

1.5.2.13 integrity

A security property of an object that prevents or is used to prevent:

- its condition of existence being changed and/or

- its contents being changed.

This property is relative to some subject population and to some agreed degree of security.

1.5.2.14 node

A data processing facility that provides information processing resources as part of a network. A node may support user application processes, server application processes or a combination of both kinds of processes.

1.5.2.15 object

Abbreviation of security object.

1.5.2.16 privilege attributes

Attributes, associated with a security subject that, when matched against control attributes of a security object are used to grant or deny access to that security object.

1.5.2.17 reference data transfer

An information transfer between two server application processes acting cooperatively on instructions from a third application process.

1.5.2.18 security administrator

An authority (a person or group of people) responsible for implementing the security policy for a security domain.

1.5.2.19 security attributes

A general term covering both privilege attributes and control attributes. The use of security attributes is defined by a security policy.

1.5.2.20 security domain

A bounded group of security objects and security subjects to which applies a single security policy executed by a single security administrator.

1.5.2.21 security facility

A set of logically associated security functions as used in the security framework.

1.5.2.22 security facility interaction

The invocation of a function of a security facility by another security facility.

1.5.2.23 security object

An entity in a passive role to which a security policy applies.

1.5.2.24 security policy

A set of rules that specify the procedures and mechanisms required to maintain the security of a system, and the security objects and the security subjects under the purview of the policy.

1.5.2.25 security subject

An entity in an active role to which a security policy applies.

1.5.2.26 server application process

An application process that implements all or part of the functionality defined by an x service definition.

1.5.2.27 service

This term is used as generic reference to distributed applications that provide "services" to other applications.

1.5.2.28 subject

Abbreviation of security subject.

1.5.2.29 supportive security application

Specific type of application that provides security services or security management capabilities at application level rather than embedded in the communications architecture.

1.5.2.30 trust (in security sense)

Confidence, that may be based on assurances which are outside the scope of this Report, that an entity to which trust is applied, will perform in a way that will not prejudice the security of the user of the system of which that entity is a part. Trust is always restricted to specific functions or ways of behaviour (e.g. "trusted to connect A to B properly"). Trust is meaningful only in the context of a security policy: an entity may be trusted in the context of one policy but untrusted in the context of another policy.

1.5.3 Acronyms

ACL	Access-control-list
ACSE	Association Control Service Element
AE	Application-entity
ASE	Application-service-element
MACF	Multiple Association Control Function
OSI	Open Systems Interconnection
PSAP	Presentation Service Access Point
ROSE	Remote Operations Service Element
SACF	Single Association Control Function
SMAE	Security Management Application Entity
SAO	Single association object
SSA	Supportive Security Application

2. REQUIREMENTS

2.1 Aims of this Report

The aims of this Technical Report are :

Stability

It shall provide a stable base for future use by ECMA standards.

ISO & CCITT

Compatibility is required with emerging work on secure distributed systems in ISO and CCITT. The proposals in this Technical Report shall be a basis for constructive ECMA contributions to ISO and to CCITT.

Upper layer mappings

The Report is limited to framework aspects in the Application Layer of OSI. It shall de-couple secure distributed systems specification from the differences between, and possible instabilities of, the various OSI "execution environments" likely to be used, and rely on the Remote Operations Notation.

Portability

The framework specification shall support a high level of abstraction, such that the specifications of secure distributed systems interactions are highly portable.

Precision

The framework specification shall be such that the specifications of secure distributed systems interactions are likely to be unambiguous.

Productivity

The recommended design principles shall be such that application designers can quickly specify, develop and deliver the interactive components of secure distributed systems.

2.2 Environment Compatibility

The Security Framework shall support control of access by subjects to applications and their protected objects independent of the location of either subject or object. It shall support the security requirements of the following types of distributed application:

- Interpersonal messaging,
- Group communication,
- Filing and Retrieval of information,
- Input and output of documents,
- Data transfer between different applications or remote servers including third party transfer,
- Directory services needed to locate and access remote applications and users,
- Time services needed for time stamping.

This list is not complete and may be extended in future.

Users and applications should be affected as little as possible by the security functions defined by this Framework. However it should be recognized that adding security functions cannot remain invisible to existing applications or operating systems.

2.3 General Security Requirements

2.3.1 User View Of Security

Users of a distributed system trust the system components to perform the expected functions and no other.

In this context, "users" include administrators, operators and owners.

These users are not solely interested in keeping information secret as is frequently emphasized in the military view of security. Certain users may be more concerned with the integrity of the resources and data of their systems. For example the integrity of a system might be compromised by "trojan horse" software which appears to act in a proper manner whilst making unauthorized use of objects which it is acting upon.

Alternatively, the integrity of a system or application may be broken by unauthorized access to the system management functions or by direct physical access. The integrity of an application used for a particular request depends on the integrity of all the system components used in supporting that particular request.

Many of the mechanisms for ensuring the integrity of a system are outside the scope of this document. However, it is possible to take steps, such as encryption and restriction of access between applications whereby if the integrity of one application is broken, it has minimal effect on other applications. It is important that such steps are taken, so as to limit the effect of any breach in security.

2.3.2 Threats to be addressed

The following subclauses describe a number of general threats to the security of of distributed systems. The threats described do not form an exhaustive list of actual threats: other threats, realized through other methods of attack may be found and used. The threats given are general enough to address most kinds of actual threats.

2.3.2.1 Disclosure of Information

Organizations maintain valuable information on their computer systems. This information may be used by other parties in such a way as to damage the interest of the organization owning the information. Therefore information stored on or processed by computer systems must be protected against disclosure both internal and external to the user organization.

2.3.2.2 Contamination of Information

This is the complement of information disclosure. Valuable information may become worthless if unauthorized information is mixed with it. The damage may be as great as the damage through information disclosure.

2.3.2.3 Unauthorized use of Resources

Generally, authenticated subjects will not be allowed to use all resources of a system. Unauthorized use of resources may lead to destruction, modification, loss of integrity etc. of resources and thus the authorization of individual users will be limited.

2.3.2.4 Misuse of Resources

Authorized use of resources may leave authorized individuals the opportunity to perform activities that are harmful to the organization. Misuse of resources, intentional or accidental, may be harmful to the organization through corruption, destruction, disclosure, loss or removal of resources. Such misuse may affect the liability of an organization for information entrusted to it or for transactions and information exchanged with other organizations.

2.3.2.5 Unauthorized Information Flow

In a distributed system information flow must be controlled not only between users of end-systems but also between end-systems. Depending on the prevailing security policy information flow restrictions may be applied on the basis of the classification of data objects and/or end-systems, user clearances, etc.

2.3.2.6 Repudiation of Information Flow

Repudiation of information flow (e.g. business communications) involves denial of transmission or receipt of messages. Since such messages may carry purchasing agreements, instructions for payment etc, the scope for criminal repudiation of such messages is considerable.

2.3.2.7 Denial of Service

Because of the wide range of services (data processing, filing and retrieval, transaction processing, etc) performed with the aid of computer systems, denial of service may significantly affect the capability of a user organization to perform its functions and to fulfill its obligations. Detection and the prevention of denial of service must be considered as part of any security policy.

2.3.3 Methods of Attack

The following subclauses describe some of the methods through which the security of a distributed system may be compromised.

2.3.3.1 Unidentified Subjects

The resources of a system must be protected from use by subjects that cannot be identified. The identification must be positive and non-forgeable. This requirement extends to both human users and applications acting as subjects.

2.3.3.2 Introduction of Unauthorized Resources

The resources of complex computer systems are subject to constant change. New software releases are introduced, databases updated and network configurations change. All these changes pose potential threats because they allow the introduction of unauthorized resources that may be used for a variety of malpractices.

2.3.3.3 Misuse of Management Facilities

Computer systems provide their users with a wide range of management facilities that control the availability, operation and performance of the system. Misuse of management facilities may have extensive impact on the capability of an organization to perform its functions and to fulfill its obligations.

2.3.3.4 Traffic Analysis

The information traffic generated by a computer system or network may offer third parties significant clues to the nature of the business being conducted. Traffic analysis may have to be discouraged or prevented.

2.4 Security Policies and Domains

2.4.1 Security Policy

To be effective, security measures need to be coherent and complete. Security measures must be defined in terms of physical security (e.g. vaults and doorlocks), procedural security (e.g. selecting personal with regard to trustworthiness, changing passwords regularly) and logical security (e.g. access controls and cryptography).

The overall set of measures is defined by a "Security Policy". Since this document is concerned with logical security, only logical security measures will be considered and "Security Policy" will be used to identify the subset of the overall security policy that relates to logical security only.

Even with this restriction security policies differ widely because of differences in the environment in which user organizations operate. For example, high value commercial transactions require security measures that are different from security measures applied to an office mail-box service used for non-confidential business mail.

Many elements of different security policies have common denominators as regards their implementation. Therefore it is feasible and advantageous to conceive a single, coherent security

framework for open systems that addresses a wide range of security policies without requiring identical implementations.

2.4.2 Security Administration Domains

Security policy and its implementation is the responsibility of the Security Administrator. The purview of a Security Administrator is known as a Security Domain.

A security domain covers all or part of a given distributed system. It is conceivable that security within a given system is the responsibility of a number of security administrators. This occurs in large international organizations where national or regional security administrators are responsible for a part of the overall company network.

Within each domain the security administrator is responsible for the implementation of the domain policy and for assuring its continued effectiveness. This responsibility includes the installation of trusted hardware and software functionality, monitoring day-to-day operations and recovery in case of breaches of security suspected or real.

Where networks are large or where local staff has a considerable degree of autonomy, the security domain may be subdivided into subdomains. In each subdomain all or part of the overall security policy applies but certain security functions (e.g. adding new users or terminals to the network) are a local responsibility borne by a subdomain security administrator.

The Security Domain concept may also be used to denote the scope of a specific set of security rules that apply, not to a network but, for example, to a single distributed application. Such an application level domain has an application level policy supported by another, wider policy that is concerned with those security aspects that are application independent.

Independently of the kind of domain considered there are different relationships between security domains.

Domains may have a peer-to-peer relationship as in the case of two company security networks.

Domains may have subdomains: the larger domain may delegate part of its policy to a subdomain. If the subdomain shares all its policy elements with the larger domain, this is known as exclusive delegation.

Domains may also have a non-exclusive relationship: an application domain may share a part of the overall policy with the overall domain and may have policy elements it does not share with the overall domain.

Within a hierarchy of domains it is important to identify where the responsibility for specific security aspects lies. Responsibility for a given object or subject must occur only once in the hierarchy.

2.4.3 Cooperation between Security Domains

By its very nature, business - whether commercial or public sector - requires the interaction between different organizations each having their own security policy. Thus doing business - even within one organization - may require the interaction between different security domains. This requires cooperation of security administrations. This requires sophisticated interdomain controls, for example with regard to access between domains.

With regard to interdomain access control a given domain may have one or more of three roles:

- authentication domain. This is the domain where a user is known and authenticable by means of information held within that domain.

- originating domain. This is the domain where a user is actually located at the time of his request for some service.

- destination domain. This is the domain where the request of the user will be serviced.

These roles may coincide or not at all. This yields the following role combinations that are relevant to interdomain cooperation:

- authentication + originating + destination role.

This is the simplest case. It does not involve interdomain cooperation at all.

- authentication + originating role.

In this case the originating domain has to properly authenticate the user and to pass trustworthy - and mutually acceptable - user identification and privilege information.

- authentication + destination role.

In this case the originating domain exchanges authentication information with the destination domain.

- origination + destination role.

In this case the authentication domain must correctly authenticate the user, and, possibly, to provide a trustworthy set of privileges.

- one role only.

This is the most complex case from an interaction point of view: the originating domain serves as a remote access service to the authentication domain which will forward authenticated requests to the destination domain. See Figure 2.

Note that a domain may delegate its authentication role to another domain. Such a delegation transaction needs to specify user ids, credentials and temporary privileges.

Special protocols are needed to perform these delegation transactions.

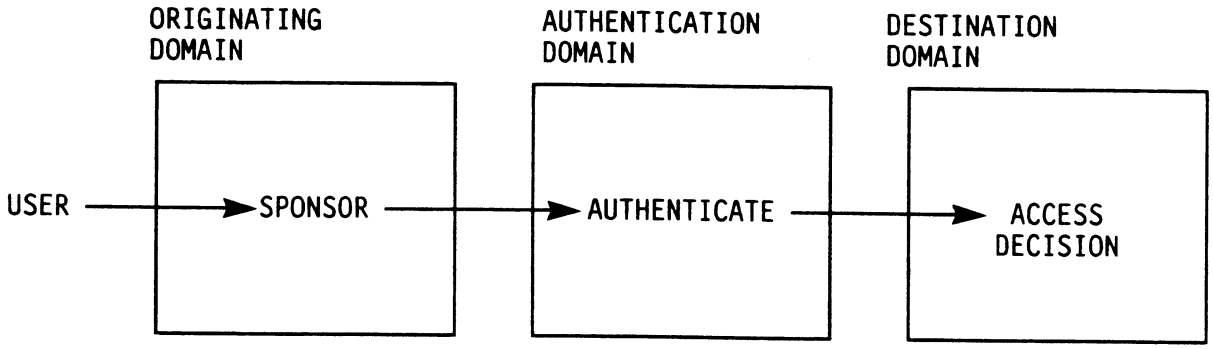


Figure 2 - Cooperation between domains with different roles

2.4.4 Levels of Policy

There are two fundamentally different levels at which the security requirements of a distributed applications network need to be addressed:

- at the application access level, concerned with access to network objects like productive and supportive applications,

- at application-specific level, concerned with access to application-specific objects like files or documents.

These two levels of view have quite different requirements, reflected in different security policy elements tailored to the different kinds of protected objects involved and the different components of the network that are responsible for their support. Indeed entities that are considered protected objects at one level can become accessing subjects at another. This is illustrated in Figure 3.

Figure 3 shows two end-users accessing two applications policed by an Application Access Policy. User 1 accesses Application 1, User 2 accesses Application2. Application 2 is shown with its internal details revealed; it is supporting objects of its own and it is itself controlling access to them via its own Local Application Policy.

The figure shows that an application (Application 1) can also take the role of an accessing subject: it accesses Application 2 and one of its objects.

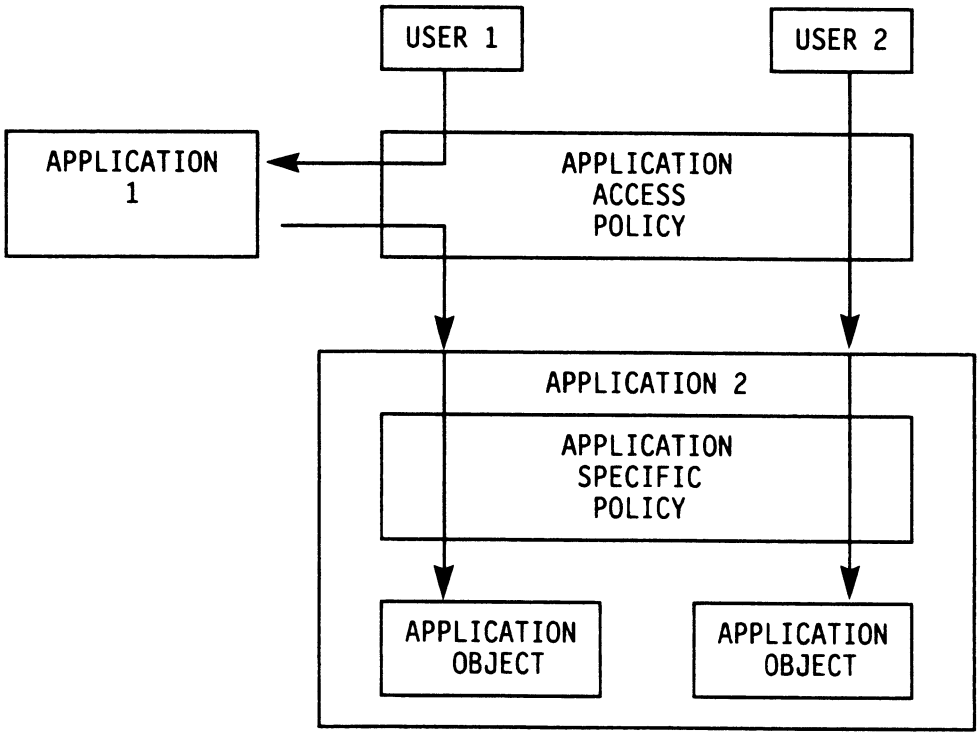


Figure 3 - Levels of Policy

2.4.5 Implementation of Policies

Security policies may be satisfied in a number of ways. If a security policy states that two groups of users will not have access to each others data then each group may be given separate computer systems placed in locked rooms. Thus, the policy is satisfied by implementing door locks. In the real world, systems are shared and applications are distributed over different systems. Therefore the control mechanisms required by a given policy will have to be distributed.

The controls themselves need to be correct and protected from subversions. In order to achieve an acceptable degree of security in terms of a given security policy, the underlying system needs to provide basic security facilities that are essentially independent of actual application security policies. It is these underlying security facilities and their interaction with other applications and their

management that are identified by this Technical Report and that should be specified as part of a security architecture for Distributed Systems.

2.4.5.1 Degree of Security

It is up to the organization responsible for a distributed system or application to choose what strength is adequate for the security functions required (e.g. password encryption) and for the types of risks to be covered (e.g. disclosure of sensitive files). The degree of security provided by a system should correspond to an accepted degree of vulnerability.

A general security architecture will provide different degrees of security which are indirectly related to different types of threats. The basic architecture should not change with the degree of security chosen.

The degree of security is related to the location of security functions in network components like workstations, Local Area Networks and Wide Area Network connections, gateways and servers.

It is implicitly related to the trust in every component of the distributed system onto which the basic security architecture is mapped.

2.5 Functional Security Requirements

The following subclauses specify security functions that serve to address the threats in Clause 2.3.2.

2.5.1 Access Control

Access control provides the means to allow only access by certain known users as well as to restrict access by these users to specific resources. Thus, access control has two major components: authentication of users and authorization of access. Access control will be exercised according to an access control policy. Such a policy defines both user authentication methods and access authorization methods and applies to what is known as a security domain. See Clause 1.5.2 for definitions of these concepts.

2.5.1.1 Authentication

Users gaining access to a distributed system will be authenticated before being allowed access to an application that is subject to an access control policy. Users may also require that the application(s) accessed are authentic.

Systems supporting user authentication must assure the integrity and, possibly, the confidentiality of the credentials involved. Authentication may be time bound; repeated authentication may be needed in certain applications.

Users may access an application from a system or terminal belonging to the same security management domain as the application or from a terminal or system belonging to another domain.

In either case a commonly agreed procedure of exchanging authentication information must be used. In order to avoid an unmanageable proliferation of procedures, standardization of both syntax and semantics for user authentication is required.

2.5.1.2 Access Authorization

Applications in a distributed services environment require the use of access authorization to protect the confidentiality and integrity of protected objects.

Authorization methods may use a variety of mechanisms such as Access Control Lists, Capabilities and other security attributes, singly or in combination.

Users will be granted access to applications and objects within applications based on their access rights under the prevailing security policy.

Whenever users access services or other objects not belonging to their domain, authorization information as required by the serving domain, will be passed to it in a secure fashion.

Such transfers will always be subject to bi-lateral agreements between the security domains involved. However, to avoid unmanageable proliferation of procedures for this type of exchange, standardization of syntax and semantics is required.

2.5.1.3 Indirect Access and Proxy

In some cases an application may be accessed by another application rather than by a human user. There are two extremes:

- the initiating application is acting on its own behalf,

- the initiating application is acting on behalf of another subject.

The first situation may be used, for example, to restrict access to objects held on one application (e.g. a File Server application) to requests coming via another (e.g. a Database application). It is entirely appropriate for the Database application to act with respect to the File Server as a subject with its own identity and access privileges. In this way, end user access to a protected object can be controlled in terms of the route and method used to access it.

The second situation occurs if a user wants to transfer a file directly from one application to another. The user requests one of the applications to initiate the transfer on his or her behalf. This application is acting as the user's proxy and must convince the security system of the other application that it is legitimately doing so before the transfer will succeed.

Hybrid cases also exist in which the initiating application uses its own privileges in combination with those given to it by the user.

Proxy mechanisms must allow access across domain boundaries.

2.5.2 Resource Protection

For the purposes of this document a "resource" is understood as any element of the collection of applications (e.g. Filing and Retrieval, Directory) and systems management functions that provide services to other, installation specific applications or serve to control and maintain the environment in which these other applications operate.

There are threats to resources not covered by access controls:

- Integrity of Resources. This addresses the threat of legitimate resources being contaminated or corrupted. Mechanisms for assuring resource integrity are not a local matter since the distribution of resources between open systems becomes visible in resource management protocols.

- Confidentiality of Use of Resources. This addresses to some extent the traffic analysis threat. There are a number of ways this can be addressed including routing control, encryption and traffic padding. Each method has protocol implications.

- Assurance of Service. This addresses the threat of denial of service. Assurance of service is an important consideration in the design of security support services. There are various ways in which assurance of service can be provided, e.g. through redundant servers. The use of redundant servers has protocol implications.

- Accountability of Usage of Resources. Users and service providers may require means to assure accountability of usage of resources e.g. for billing. Accountability includes the selective logging

of an audit trail of operations, attempted or completed, as well as non-repudiation of data origin and of receipt.

Auditing is treated more fully in Clause 2.5.4.2 "Security Event Detection".

Non-repudiation is proof, a posteriori, to a third party, of the identity of an entity that sent or received a given message.

2.5.3 Information Protection

For the purposes of this document "information" is understood to refer to all information other than that needed for the control and maintenance of the distributed computer system. Thus "information" typically refers to user generated information and information processed on user instructions.

Users and service providers may require the protection of information, interchanged with or stored on a distributed services system, from external attack. Information protection covers both confidentiality of existence and content and, integrity of existence and content.

Within a Distributed Systems environment the following requirements with regard to data protection apply:

- Protection during processing,
- Protection in storage,
- Protection in interchange,
- Protection of information flow.

The information involved may be user files, programs, documents,messages, etc. as well as system owned objects such as programs and configuration data.

Unless physical protection is to be depended upon, confidentiality and integrity require the use of cryptographic techniques. These techniques require the use of cryptographic keys.

Systems supporting key distribution must provide a secure method of managing keys both within a domain and between domains.

2.5.3.1 Protection during Processing

This requirement applies only to end-systems and is therefore outside the scope of this document.

2.5.3.2 Protection in Storage

The use of encryption within any component of a distributed services system is a local concern that falls outside the scope of this document. However, in order to avoid proliferation of encryption techniques, standardization is required for their use in interchange. This includes the encryption of information in storage.

2.5.3.3 Protection in Interchange

Protection in Interchange may take various forms:

- Interchange Confidentiality. This addresses the need to keep the existence and/or the content of certain interchanged confidential,
- Interchange Integrity.This addresses the need to assure the integrity of interchange data,
- Non-Repudiation of Origin. This addresses the capability to assess the liability of the originator of an interchange,

- Non-repudiation of Receipt. This addresses the capability to assess the liability of the accepter of an interchange,

Non-repudiation may involve a third party - trusted by both the other parties - directly in which case a protocol between all three is needed, or indirectly.

Non-repudiation is closely related to notarisation. In notarisation of a transaction, a trusted third party, the notary, "seals" information shared by clients of the notary.

Interchange security requires the use of cryptographic techniques.

2.5.3.4 Protection of Information Flow

Information Flow Control complements data protection in that it assures that data which has a given security or integrity classification is not made available to processes of lower security or higher integrity classification. Information Flow Control may require the use of data labelling in such a way that the security (or integrity) label is bound securely to the labelled data.

2.5.4 Security Management

Secure Distributed Systems provide the operating organization with the tools to install, operate and maintain the security facilities and security services of these systems. Examples of these tools are secure software installation, secure channels for the transport of systems information and audit facilities for auditing the operation of the security facilities and services.

For every kind of security function three aspects of management should be addressed: administration, detection, and recovery. Depending on the degree of security desired some or all of these aspects become visible in actual implementations.

2.5.4.1 Security Administration

Security Administration has two aspects:

- gathering security management information about the system

- entering security management information into the system

The first aspect concerns reports made from information logged by the audit facilities.

The second aspect deals with the entering, addition, changing or the deletion of subjects and objects and with the definition of identities, credentials, rights, cryptographic keys, etc.

Distributed Systems provide the tools for their administration. There may be several administrators at various places in a network. In addition, certain functions related to the administration may, in some cases, be performed directly by special users and not by administrators (for example: attribution of rights over a resource may be performed directly by the owner or a resource and not by an administrator).

The total security administration of a Distributed System may thus be a complex operation that requires frequent interactions throughout the system.

Specific protocols to support this are for further study.

2.5.4.2 Security Event Detection

Detection is based upon auditing of the security operations of a system.

Auditing of security related operations provides system administrators with feedback concerning the use and effectiveness of the security functions of the system.

Auditing has four components:

- audit trail content specification

- audit trail analysis,

- audit reporting, and

- audit trail archiving.

The last belongs to the aspect of Administration.

Generation and collection of audit trail information occurs on each security event e.g. sign-on of a user, access to a protected resource establishment of a new crypto key. This information is collected according to defined specifications. In effect, these specifications provide a level of filtering. In a distributed system support must be provided for exchange of such audit content specifications.

In large systems the amount of audit information may be quite high. Audit trail analysis sifts the audit information with regard to significance, exception conditions, etc. Real-time analysis allows the early detection of (potential) security violations. This real-time analysis may be a distributed function in itself and thus protocols are needed to carry alarm conditions detected in many different places for reporting and decision making by security management.

In a distributed systems environment an application may be distributed over multiple domains. Where such is the case commonly agreed audit techniques may be needed to facilitate inter-domain cooperation and thus a measure of standardization is desirable.

2.5.4.3 Security Recovery

Recovery of a security breach - real or suspected - may require changes in security procedures and information available at the different nodes of a distributed system. Therefore protocols and procedures are needed to support the implementation of recovery measures.

2.6 Implementation Considerations

The following considerations impose constraints on the implementation of security functionality from the viewpoint of existing architectures and currently used technologies.

2.6.1 Use of Supportive Applications

The distributed applications environment (ECMA TR/42) provides a number of supportive applications which may be used effectively for security related functions provided that those supportive applications can be trusted or that no trust needs to be placed on them.

On the other hand these same supportive applications may be policed objects under the prevailing security policy.

2.6.1.1 Directory Application

Directories provide their users with information regarding the names and location of resources and users as well as with some types of attributes related to them. Directories exists in various contexts and at different levels e.g. public directories and private directories. The difference between actual directories is not necessarily functional; in fact, for reasons of interoperability, compatibility between public directories operated by service providers and private directories operated by individual users is highly desirable. Instead, the differences are more likely to be found in the type of information contained in them and the level of trust they are assumed to warrant. For the purposes of this document a directory is understood to be a private directory that is subject to the same security policy as the other resources of a given security domain.

This private directory may be used to hold a variety of information including security attributes of subjects and objects that belong to a given domain. Thus such a directory may be used in the

process of user authentication and security attribute management. This subject is further explored in Clauses 4.2 and 4.6.

The use of private Directories that conform to the functional standards laid down for public directories has implications in terms of the naming conventions these standards impose on user organizations. The security consequences of these naming conventions are for further study.

2.6.1.2 Time Application

A variety of security functions need a reliable timing service e.g. for time stamping transactions or audit trails. Where applicable, security functions will make use of the Time Application.

2.6.2 Cryptography

Many security functions rely on cryptographic processes to assure confidentiality and integrity. Interoperability between Open Systems using cryptography requires:

- the use of agreed algorithms,
- the negotiation of algorithm and mode of operation during interchange,
- the use of key management procedures and protocols.

Selection of algorithms is outside the scope of this document.

2.7 Design Requirements

The following requirements constrain the design of trusted systems and apply to the design of this Security Framework.

2.7.1 Separation of Functionality

A central tenet of computer security theory holds that the "controlled" should not "control" e.g. users subject to an access control policy should not have access to the mechanisms and the data that implement that policy. Thus users for whom access control attributes are held in a directory should not be able to change the content of that directory so as to increase or change their own or other's access rights. Changing the directory content should be done by "users" - security administrators - with special authorization to do so. The security administrators on the other hand, should not be able to give themselves access rights to user owned resources and objects.

A further implication is that security functions should not depend completely on the trustworthiness of the resources and facilities they use.

2.7.2 Distributed Operation

In a distributed applications environment not only the resources and users are distributed, also the security "services" themselves are distributed. The functionality of these services should be location independent and allow of distributed operation

2.7.3 Robustness/Resilience

Security should be robust or resilient in the sense that malfunction or failure of individual system components does not endanger the security of the system components and applications remaining active.

Also, any breach in security should have local effects only that can be remedied by adequate recovery actions. In high security systems the dependence of security "services" on the availability of resources shared with users has to be avoided or minimized.

2.7.4 Selective Implementation

An important constraint on the security architecture is that it should allow selective implementations: not all user requirements are alike. Further, users not requiring any security in their systems should not be confronted with constraints that might result from security implementations they do not need or use.

2.7.5 Usability

Adding security to a system adds to its functionality and will impact the way users get access to the system and it will impact the performance of the system. By suitable design of security functionality and security protocols these impacts must be minimized. The extent to which this is possible depends on the security policy to be supported.

2.7.6 Evaluation and Testing

No system should be trusted if its design cannot be evaluated and if its security performance cannot be tested. Thus security functions and protocols should be kept simple and compact to allow evaluation and testing, both of which may be done by organizations other than the systems vendor or the user. Accepted standards for commercial secure systems evaluation and testing are not available yet. This Framework may contribute to the development of such a generally accepted standard.

2.7.7 Certification and Accreditation

Successful evaluation and testing may result in the certification of a secure system. On the basis of such a certification individual user organizations may accredit their systems relative to the security requirements established by their security policy. This subject is outside the scope of this Framework.

3. SECURITY CONCEPTS AND MODELS

This Clause develops the two main concepts used in the ECMA Security Framework: the Security Domain and the Security Facility. These concepts can be mapped to the OSI Reference Model and they can be applied to modeling distributed security applications and distributed security management.

3.1 The Security Domain Concept

3.1.1 Introduction

The security domain is a managerial concept that limits the scope of a particular security policy. Thus a domain will cover a number of users (subjects) and computing resources (objects). Where the number of subjects and objects is large,they are often formed into subgroups for ease of management. If a subgroup has a policy different from the policy of the whole, the subgroup is referred to as a subdomain. Normally the policy of the overall domain will apply to all subdomains. In addition each subdomain may have freedom to interpret some aspects of its security policy to meet local needs. The concept of a subdomain is only used to indicate a certain relationship between two domains. Such a relationship may exist at multiple levels: a subdomain may have another subdomain below it.

There are two basic types of relationship between security domains:

- the subdomain relationship. This is described above. In Figure 4, domain A.B has a subdomain relationship with domain A.
- the peer relationship. Two domains have a peer relationship if neither domain is a subdomain of the other. In Figure 4, domain A.B and A.C are peer domains. Peer domains may share a higher

common domain (domain A in Figure 4) that may provide facilities for interaction between the two peer (sub-)domains. An example would be an Authentication facility shares by A, A.B and A.C. The way domains and subdomains interact will be specified by the relevant security policies.

A special kind of peer relationship exists where two domains do not have a common higher security domain. Such peer domains may be called "autonomous peer domains". In Figure 4 the peers of domain P are domains A, A.B and A.C. For interworking between autonomous peer domains, special measures are required that are subject of the security policies of the domains involved. Note that the interdomain elements of the policy for domain A.B may not allow direct interaction with domain P but require that such interactions involve domain A.

3.1.2 Autonomous Peer Domains

The simplest relation between two domains is the peer-to-peer relationship. Each domain is considered autonomous to the other; each has its own policy and there is no common, controlling policy which may be invoked in communications between the two domains. One could define the peer relation between domains as the absence of a higher domain that polices their interaction.

In the real world interactions are frequently peer interactions between people, between departments, between companies, etc. each of which can be understood as a separate security domain. In view of the importance of peer interaction between security domains, this Security Framework provides a specific security facility to model such interactions.

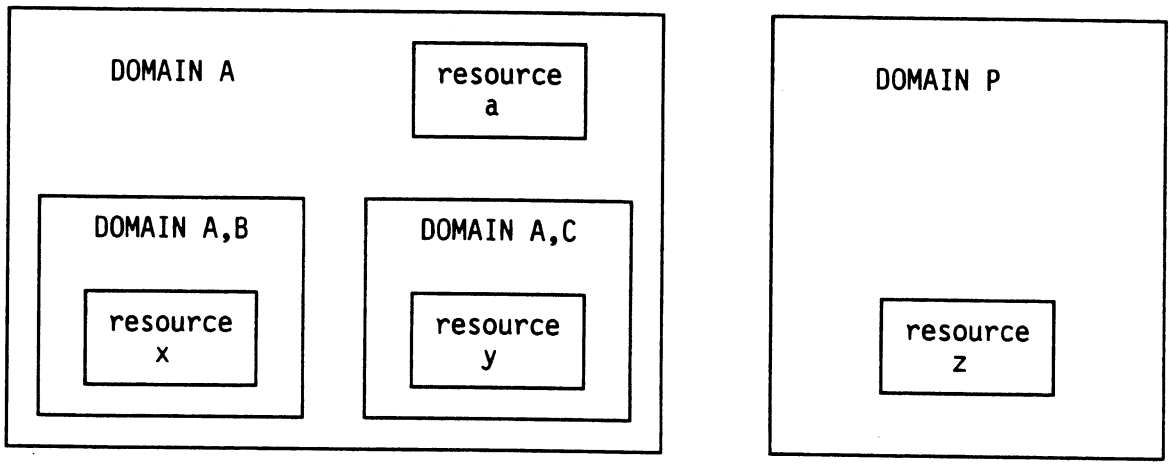


Figure 4 - Relationships between Security Domains

3.1.3 The Security Subdomain

The relationship between domains A and A.B and A.C is one of domain to subdomains. This is reflected by the notation. The mapping between the policy of domain A and its subdomains is important, it determines the way resource x and y may be accessed. There are two policy relations which reflect the ends of a spectrum of possible mappings:

- Non-exclusive delegation. The subdomain acts autonomously on all policy matters, except those dictated by the higher domain. The freedom to act on shared policy matters is delegated without restricting the subdomain in implementing its own policy matters.
- Exclusive delegation. The subdomain acts autonomously only on policy matters delegated by the

higher domain. There is no freedom do implement local policy matters independently of the higher domain.

Where a subdomain acts autonomously on some aspects of security policy it is to all intents and purposes a security domain for those aspects. A domain may delegate the ability to manage security policy to a subdomain but still keep control of all or some of the enforcement mechanisms (the security facilities). The subdomain relationship indicates that a subdomain manages some of its own affairs, and that some others will be managed by the higher domain. For example, the higher domain may control subject access to the subdomain but leave the subdomain the freedom to decide which of these subjects are allowed access to the objects of the subdomain.

Variations of these schemes allow complex security policies that cover multiple subdomains to be accommodated.

3.1.3.1 Hierarchy and Inheritance

There are two advantages to the subdomain concept. The first is that a hierarchic structure enables the higher domain to arbitrate communication between entities in different subdomains; inter-subdomain communication can be as secure as intra-domain communication to the level allowed by the higher domain. (Thus the higher domain may act as a trusted intermediary for its subdomains with respect to interactions between themselves and between the subdomain and another domain at peer level with it). This advantage is invoked at the time of use of communications facilities. The second advantage is the idea of inheritance (which may be seen as the mirror image of delegation). A subdomain inherits some of its security policy from the higher domain. This advantage is a static management feature that helps to simplify inter-domain operations.

Assume that domain A (in Figure 4) has a security policy A:= < STATEMENT 1, STATEMENT 2 ... >. Domain A.B will inherit this policy and add its own statements <STATEMENT α, ...> so that is policy becomes A.B:= <STATEMENT 1, STATEMENT 2 ... STATEMENT α, ...>.

A real example is the context of audit policy may be as follows:

A:= <, ALL SECURITY EVENTS SHALL BE LOGGED IN JOURNAL FILE AUDIT, ... >.

AB:= <, ALL SECURITY EVENTS SHALL BE LOGGED IN JOURNAL FILE AUDIT AND LOGGED ON PRINTER P, ... >.

Policy inheritance between subdomains is also important when communication between resources in different subdomains is required. If both subdomains have inherited a policy that allows inter subdomain communication (which must be supported by a suitable security mechanism) then the resources in the subdomains can utilize this policy to obtain secure communication without explicitly negotiating a trust relationship. However, if the subdomains inherit no policy on inter-domain communication (the policy inheritance was non-exclusive with regard to inter-domain communication) then A.B and A.C appear as autonomous domains and must interact as peer domains.

Policy inheritance will affect the security information that Open Systems need to exchange in interdomain access. It also affects the mechanisms needed to allow joint control by Security Administrators over attributes that may have relevance in more than one subdomain.

3.1.3.2 Some Examples

Both examples use Figure 4 as graphic illustration.

Distributed System Example

In this example, the overall domain (A) is a company network.; the subdomains are departmental networks. The resources x and y are workstations on the departmental networks that wish to communicate. Depending on where authentication is performed this example can be used to demonstrate two cases:

- In the first case domain A supports an authentication service (resource a in the Figure) which is used by all subdomains. Although the authentication facility is not delegated, the managers of the subdomain may still have the authority to manipulate the authentication database in line with their own local policy.
- In the second case each subdomain is assumed to have its own authentication service. If the two workstations need to communicate the need the services of domain A which will guarantee each service to the other.

In both cases the security of the inter-domain communication depends on the degree of security provided by the higher domain.

Service Bureau Example

For this example lets assume that Domain A covers a time-sharing computer on which a bureau service is being run. The bureau has two customers, who may be competitors. In this example it is very important that interaction between the domains is strictly controlled, if not forbidden. The bureau management would normally delegate a number of facilities to the customers, for instance: creating users, file protection between local users. The bureau would not delegate file management that would allow one customer to access the files of another. If a user x wanted to communicate with a user y (say by some messaging facility) then they would have to communicate through the higher domain. The ability to communicate directly would not be delegated. This is an example where the sub-domain would only have freedom explicitly allocated.

From the above examples we can see that the simple sub-domain concept is quite powerful and covers a wide range of situations. The important simplifying concept is that each domain is treated as a separate autonomous domain unless it is useful, or necessary to consider it as a sub-domain. The main point is to avoid complex hierarchic relationships which cannot be implemented in the real world.

3.1.4 Types of Security Domain

Since the concept of a security domain is closely related to the scope of a security policy there can be many types of domain. The following are three examples (see Figure 5):

- The distributed system security domain. Its scope is the security of interworking between end-systems. The distributed system policy and end-system policy overlap where both rely on the same trusted functionality. Generally the former policy will provide the binding factors that tie application policies and end-systems policies together into a coherent security system.
- The end-system domain. Its scope is the individual end-system, including its hardware based security functions which are the basis of each trusted system. In practice many end-system domains are subdomains with exclusive policy delegation from a common distributed system security domain.
- The application security domain. Its scope is a given application e.g. a directory, a filing and retrieval service, etc. An application security domain may overlap a number of end-system domains. Also, multiple applications may reside on one end-system. In practice, application domains are subdomains with non-exclusive delegation of policy from the end-system or dis-

tributed systems security domain. Application domains will generally be peer domains for each other.

Policy relations and domain types play an important role in the actual management of secure systems.

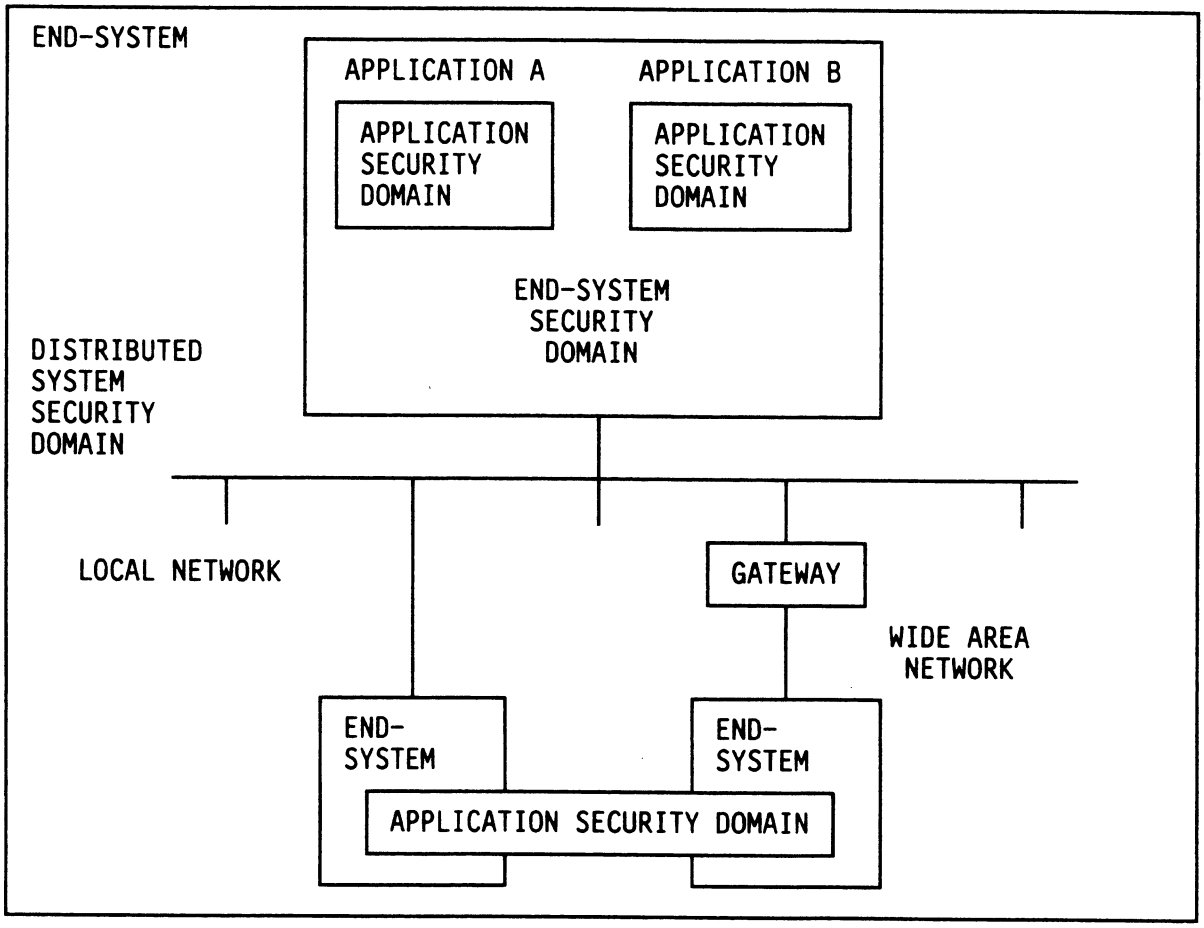


Figure 5 - Types of Security Domains

3.2 The Security Facility Concept

3.2.1 Introduction

Security in a distributed system requires the implementation of many different functions. These functions, which are modeled as Security Facilities, are described in detail in Clause 4, a walk-through is given in Clause 4.10.6.

The security facilities are a modeling technique for developing and describing a framework for Security in Open Systems. The facilities and their interactions are logical entities within the framework; they are not designed to represent, nor describe, the implementation of a secure system. The framework is designed to provide a model from which a comprehensive and complete set of security services and protocols may be developed. An implementation of an Open Secure System will make use of these services and protocols to support a specific security policy.

The following Clauses describe a functional decomposition of "security" into ten security facilities:

Subject Sponsor Facility
Authentication Facility
Association Management Facility
Security State Facility
Security Attributes Management Facility
Authorization Facility
Interdomain Facility
Security Audit Facility
Security Recovery Facility
Cryptographic Support Facility

These facilities will be used as building blocks of Supportive Security Applications for use in distributed systems. See also Clause 6.

No assumptions should be made about the possible distribution of the security facilities over the components of a network. This might vary from being a single central network service that implements all facilities to being an aspect of every distributed supportive or productive application.

3.2.1.1 Overview

Subject Sponsor

The Subject Sponsor is the intermediary between the security subject and the other security facilities. It is the only entity in a distributed system that is aware of all of a subject's current and sometimes concurrent activities while accessing protected objects or applications. The subject involved will usually be a human end user, but under network policies where services' access to other services is policed, the subject can be an accessing application.

Authentication Facility

This facility accepts and checks subject authentication information, communicating its conclusions to other Security Facilities as needed under the prevailing policy. The subject involved will either be an end-user via his Subject Sponsor or a service or application acting as a subject.

Association Management Facility

This facility provides the following kinds of association security:

- authorization, via an Authorization Facility, for the two entities to communicate,

- assurance of the identity of the communicating entities. This assurance may be provided explicitly to each entity or it may be implicit i.e. based on a trust relationship between an entity and, say, the Authentication Facility.

- control over the kind and type of security of the underlying communications service e.g. by enforcing routing of a connection such that only trusted links are used.

Security State Facility

This Facility maintains a view of the current security state of the system in terms of authenticated subjects and objects in the system, their associations and the security attributes carried by those associations. This view is instantaneous in the sense that it reflects the security relationships within a given security domain. All other Security Facilities communicate the results of their actions to the Security State Facility. Another way of viewing Security State is as an abstraction of the state information maintained by all Security Facilities.

Security Attribute Management Facility

This Facility provides for the creation, distribution, revocation, archiving and destruction of security attributes of subjects and objects within a given security domain:

- subject-related access privilege attributes for known subjects, which may be human subjects or services and applications in an active role.

- object related access control attributes for protected objects, which may be services or applications,

- objects related access control attributes for objects belonging to a particular service or application.

The attributes may vary from Capabilities issued to users to, at the other extreme of the spectrum of security attributes, Access Control Lists.

The scope of these attributes is limited to the Security Domain to which the Facility belongs.

Subject attribute management is closely related to authentication and the two are often considered as one single facility, particularly since the privilege attributes granted to a subject might be affected by the authentication route used. There is an important distinction between the two Facilities, which is necessary when different authorities are responsible for the introduction of users and the granting of privilege attributes.

Authorization Facility

This Facility uses access context, subject access privilege attributes and object access control attributes to authorize or deny requested accesses by subjects to objects. It embodies the mechanisms needed to obtain access control decisions on the basis of the attributes presented to it. It should be noted that different types of mechanisms may be used for different levels of access, e.g. authorization of access to an Application may use a Capability based mechanism whereas access to objects supported by that Application may be policed using Access Control Lists.

Inter-domain Facility

This Facility maps one domain's interpretation of security attributes (subject identity, object identity, authentication and authorization data) into another domain's interpretation. It helps Association Management form associations between entities in different domains.

Security Audit Facility

This Facility receives event information from other Security Facilities for recording and/or analysis. Security Audit may operate at different levels: at network or application access level as well as at intra-application level. The difference is in the scope, not in the functionality.

Analysis of audited events may be done in real-time so as to detect possible breaches of security. Such analysis may be passed to a Security Administrator or to the Recovery Facility for action.

Security Recovery Facility

This Facility acts either on information received from the Security Audit Facility according to a set of rules defined by a Security Administrator or on information provided directly by the Security Administrator. For example, too many bad passwords from a given terminal may result in the terminal or the user identity under attack being locked out pending management action.

Cryptographic Support Facility

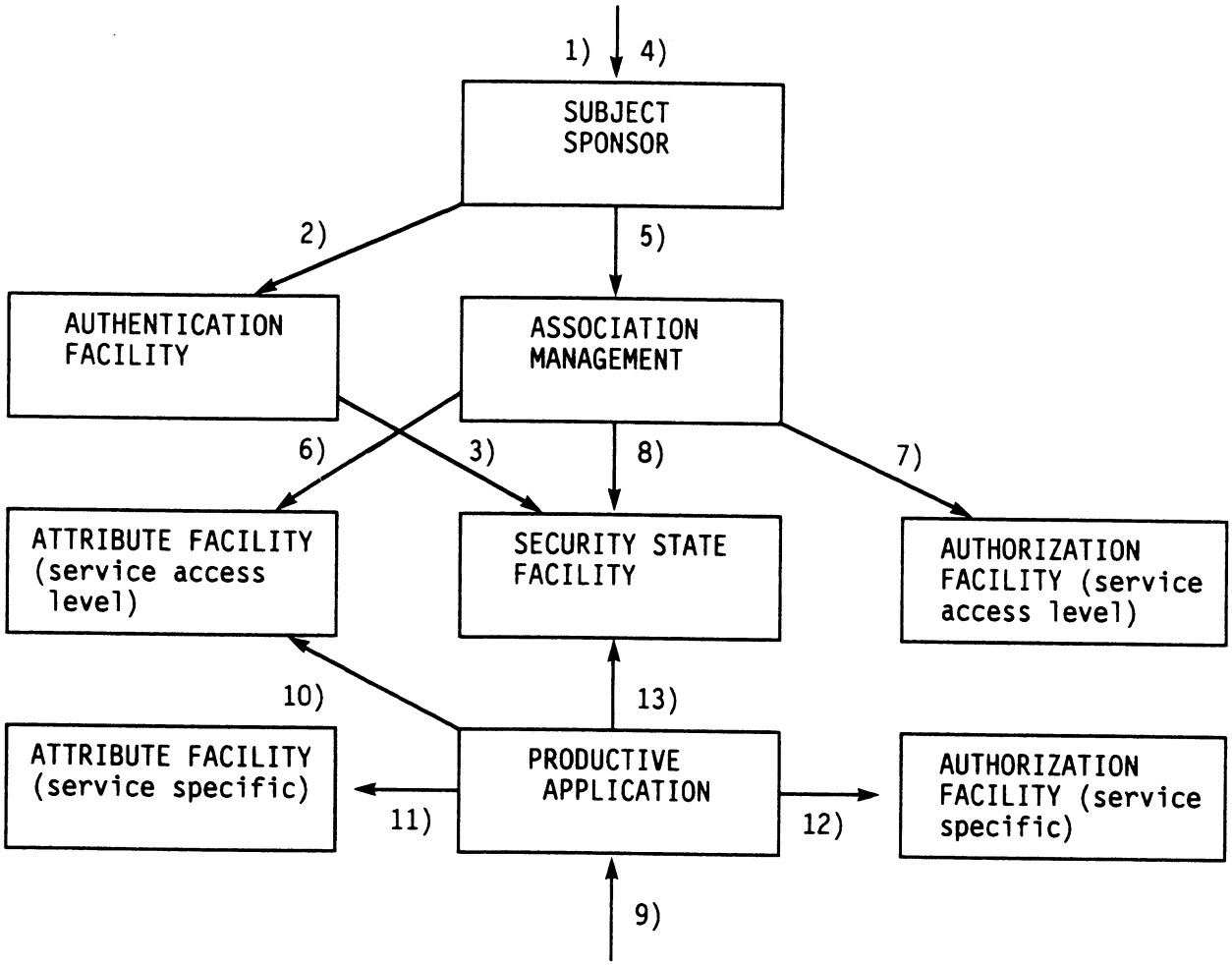
This Facility provides cryptographic services used by other Security Facilities and by productive services and applications to secure data in storage and transit.

3.2.1.2 Walkthrough

The following gives an example walkthrough of a user accessing an object controlled and supported by a productive application. The walkthrough is provided for illustrative purposes only; not all Security Facilities are involved.

At the beginning of this walkthrough the single domain system's security services which contain the security facilities are assumed to be installed and active. The productive application of the example is already in service and authenticated as an accessible security object.

Assume a user wants to update a file which is owned by a given application. First the file has to be opened. See Figure 6. Note that none of the interactions with the Security Audit Facility are shown so as to keep the Figure simple. The description assumes a successful operation from beginning to end.



Legend : A ------> B = A invokes B

Figure 6 - Security Facilities Walkthrough

1. The user activates the terminal. The Subject Sponsor is activated and requests the user's identification and authentication data,
2. The Subject Sponsor passes this information to the Authentication Facility.
3. The Authentication Facility performs the authentication using the credentials. Security State is updated with the authenticated user ID (i.e. the ID under which that user is known within the security domain).
4. The Subject Sponsor requests and gets the application name from the user.
5. The Subject Sponsor prompts Association Management to perform the association with the requested application.
6. The Association Management Facility requests the security attributes of the user and the application from the Security Attributes Facility.
7. The Association Management Facility passes user and application attributes to the Authorization Facility and requests an access control decision. A "secure association" is set up between the user's terminal and the requested application.

8. Security State is updated with the status of the association.
9. The user passes a request to the application e.g. "OPEN FILE".
10. The application obtains the user's security attributes from Security Attribute Management Facility.
11. The application obtains the security attributes of the requested file from the (its local) Security Attribute Management Facility.
12. The application presents the user's attributes and file attributes to the (local) Authorization Facility to obtain an access control decision. The user is allowed access to the file.
13. Security State is updated.

4. DETAILED DESCRIPTION OF SECURITY FACILITIES

This Clause describes the Security Facilities, their purpose, their functions and their interactions in detail.

The description follows this pattern:

- introduction on the use and purpose of the facility,
- functional description,
- interactions with other facilities,
- interactions with layer management,
- use of other applications
- facility management requirements.

With regard to the interactions described, the following should be kept in mind:

- interactions are described only with the invoking facility,
- the syntax of these interactions is irrelevant at this level of description since no assumptions are made about the environment in which this security model is applied.

4.1 Subject Sponsor

4.1.1 Introduction

When a user logs on to a distributed system using the Authentication Facility, there is a requirement for local functionality that sponsors that subject to the system during authentication and which monitors subsequent activity.

A similar requirement exists when an application or service is installed and is to be made available to users and other applications.

4.1.2 Functionality

The Subject Sponsor Facility provides all or some of the following functionality as required by the prevailing security policy:

- it acts as intermediary between the subject and the Authentication Facility using the appropriate authentication protocol,

- it interacts with the Association Management Facility for the purpose of establishing secure associations between the subject and the application(s) he wants to access,
- it monitors the presence and activity of the subject and takes action (e.g. warning of the subject, initiation of re-authentication, termination of association) as specified by policy after a specified period of inactivity has elapsed,
- it supports re-authentication of the subject during established associations,
- it interacts with the Security State Facility to record changes in the authentication status of the subject and changes in the associations established on behalf of the subject,
- it interacts with the Security Audit Facility to record all security events specific to its operations and actions,
- it may interact with the Cryptographic Support Facility e.g. for the encryption of a human user's password for protection during communication,
- it responds to security recovery information from the Security Recovery Facility.

4.1.3 Interaction With Other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE SUBJECT SPONSOR		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED/RECEIVED
AUTHENTICATION	SUBJECT AUTHENTICATION OPTIONAL ELEMENTS	USER IDENTITY, AUTHENTICATION INFORMATION / AUTHENTICATION RESULT CHALLENGE TO USER/ RESPONSE CHALLENGE FROM USER/ RESPONSE
ASSOC.MAN'T	REQUEST FOR ASSOCIATION	USER ID, RESOURCE ID, USER SUPPLIED PARAMETER/ ASSOCIATION STATUS
SECURITY STATE	CHANGE STATE	AUTHENTICATED USER ID/ASSOCIATION STATE, AUTHENTICATION STATUS
ATTRIBUTE MAN'T	NONE	
AUTHORIZATION	NONE	
INTERDOMAIN	NONE	
SECURITY AUDIT	AUDIT	AUDITABLE EVENTS
SEC. RECOVERY	NONE	
CRYPTO. SUPPORT	ENCRYPT/DECRYPT	AUTHENTICATION INFORMATION, CHALLENGE PARAMETERS

4.1.4 Interactions with Communications Layer Management

None.

4.1.5 Use of Other Applications

None.

4.1.6 Facility Management

The Subject Sponsor Facility requires the following management:

- setting/changing of policy parameters e.g. timer values, actions to be taken, etc
- set/change/delete auditable event rules

4.1.7 Characteristics of the Subject Sponsor

This Subclause is included to illustrate the nature of the Subject Sponsor in real systems.

The Subject Sponsor is the point of contact between a subject and a security domain. In distributed systems the Subject Sponsor may be implemented in different ways depending on how trust is distributed and how the connection between the subject and the secure system is made:

- if access is made through a workstation that belongs to the domain, the Subject Sponsor is located in the workstation,
- if access is made through a dumb terminal attached to a mainframe then the Subject Sponsor is mainly located in the terminal handler of the mainframe.
- if access is made through a portable PC using a dial-up telephone line to access a secure system, the Subject Sponsor may be located partially in the PC and partially in the system.
- if the subject is a human being using a hand-held security device such as a smart card or a cryptographic authenticator, these devices may take on some of the Subject Sponsor role and they may be able to provide a trusted authentication path through an untrusted intermediary - e.g. the untrusted portable PC. The other functions of the Subject Sponsor would be located within the trusted system being accessed.
- if the subject is an application or part of a distributed application acting as a subject and it wants to access another application, it is sponsored by the operating system of the end-system on which it resides.

  The same applies when an application or part of an application is coming on-line and making itself available.
- it is possible to model an operating system as an application in OSI terms. In this view it is sponsored by one of its parts, typically a software distribution/installation function that announces that the operating system is present and available. This is further discussed in Clause 7.2. If an operating system is modelled this way, its sponsoring of applications can be modelled in a recursive fashion.

4.2 Authentication Facility

4.2.1 Introduction

The Authentication Facility provides for the authentication of subjects e.g. human operators as well as applications. In addition the Facility may provide users with assurance of its own identity.

Authentication requires the use of credentials to verify authentication information supplied by a subject. Depending on the trust relationships involved, different authentication mechanisms may be used.

4.2.2 Functions Of the Authentication Facility

In general, authentication relies on an entity proving its identity by showing that it is in possession of some piece of secret information. This takes three basic forms.

a) The entity produces a copy of a piece of information that it keeps secret and that the facility associates with that entity, e.g. the classical password approach.

b) The entity uses some piece of information without passing it explicitly such that the facility, which also knows it, can prove to its satisfaction that the sender was in possession of the information, e.g. cryptographic techniques based on symmetrical key schemes.

c) The entity uses some piece of information without passing it explicitly such that the facility, which does not know it, can prove to its satisfaction by the use of some shared piece of information that the sender was in possession of the information, e.g. cryptographic techniques based on asymmetrical key schemes.

Whichever technique is used, there are three important characteristics.

1) Authentication is an end-to-end operation between the two entities concerned.

2) The result of authentication is a mapping between a communicating entity (the subject) and a verified identity.

3) The authentication result is a proof of identity at an instant in time. Assurance of the continued validity of the mapping of this identity is to be provided by other means.

The other functions of the Authentication Facility are:

- In a multi-domain environment it interacts with the Interdomain Facility. See Subclause 4.7.
- It interacts with the Security State Facility to record the authentication status of the subject,
- it interacts with the Security Audit Facility to record all security events specific to its operations and actions,
- it may interact with the Cryptographic Support Facility e.g. for the encryption of a human user's password for protection during communication,
- it responds to security recovery information from the Security Recovery Facility.

4.2.3 Interactions With other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE AUTHENTICATION FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED/RECEIVED
SUBJ. SPONSOR	NONE	
ASSOC. MAN'T	NONE	
SECURITY STATE	CHANGE STATE	AUTHENTICATED USER ID, STATUS
ATTRIBUTE MAN	NONE	
AUTHORIZATION	NONE	
INTERDOMAIN	REMOTE AUTHENTICATION	SUBJECT ID, AUTHENTICATION INFORMATION, ETC.
SECURITY AUDIT	AUDIT	AUDITABLE EVENTS
SEC. RECOVERY	NONE	
CRYPTO SUPPORT	ENCRYPT/DECRYPT ETC.	USER SUPPLIED INFORMATION, OTHER PARAMETERS

4.2.4 Interactions with Communications Layer Management

None

4.2.5 Use of Other Applications

The Authentication Facility may use a Directory Application to obtain subject related information

The Authentication Facility may use a Time Application to time-stamp its operations and their results.

4.2.6 Facility Management

The Authentication Facility supports security management functions as required by a given security policy, e.g.:

- create (=enter) credentials for ID x,
- delete credentials of ID x,
- change credentials for ID x,
- set/change/delete credentials change date
- suspend credentials of ID x
- set/change/delete auditable event rules.

4.3 Association Management Facility

4.3.1 Introduction

When entities of a distributed system exchange information, an association exists between them e.g. between a client and a server. In order that the service can be provided in a secure manner it is necessary that the association between the entities is set up and maintained securely.

This association may not map to one-to-one on the A-Associate of the OSI architecture. Depending on the environment, different mappings to real communications connections may occur: in a Transaction Processing system multiple dialogues with different security attributes may use the same underlying "association".

The entities involved may vary with the environment and the security policy: it may be a Transport Entity or, for example, an application process such as the Message Transfer Agent of an X.400 network. The latter example raises the possibility that the application involved in a given association may not be the "final destination". End-to-end security in terms of associated Application Processes is too simple a view.

Each association must be secure from three points of view:

- it must be authorized,
- the entities must have been authenticated,
- the communications channel must be secure.

4.3.2 Functions of Association Management

Secure management of associations requires:

- selection of the remote entity or application

  Generally, selection of the service or resource to be accessed is straightforward: the subject knows the name or an alias of the desired object. However, under some policies, not all these objects may be accessible from a given end-system and subjects may not be allowed to know which applications and resource can be accessed from a given end- system. Where a subject is allowed to select resources from e.g. a menu, Association Management - with the assistance of the Authorization Facility - will provide filtering of such menus in accordance with the security attributes of the subject and of the applications and resources.

- assurance that the access is authorized

  Once a subject has selected an application or resource, Association Management assures that the requested access is authorized. It does so with the aid of the Authorization Facility to which it presents the security attributes of the subject and the security attributes of the selected application or resource.

- assurance of the identity of the communicating entities

  Under some security policies authentication of the entities involved in an association may be required. The Association Management Facility will provide the services needed to allow the authentication of the entities involved.

  It should be noted that these services are needed only where explicit authentication of the entities is needed.

  The mechanisms needed for this authentication process may be supported at lower layers of the communications architecture (See ISO 7498/2).

- assurance that the underlying communications are secure

Security attributes of subject and or selected applications and resource may specify the need for secure communications to be used between the two. Association Management will assure the setting of the appropriate Security Service parameters for the association.

For associations within a security domain, the Association Management Facility selects parameter values according to the security policy of the domain. For associations between domains the Association Management Facility interacts with the Inter-domain Facility to establish parameter values appropriate to the policies of both domains.

Secure communication may be achieved in a number of ways including the use of cryptography by the associated entities, use of cryptography at lower layers or through the selection of secure links over which the connection is routed (See ISO 7498/2).

4.3.3 Interaction With Other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE ASSOCIATION MANAGEMENT FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED/RECEIVED
SUBJ. SPONSOR	NONE	
AUTHENTICATION	NONE	
SECURITY STATE	READ STATE	E.G. SUBJECT STATE, OBJECT STATE
	CHANGE STATE	E.G. ASSOCIATION NAME, SUBJECT AND OBJECT STATE, ETC
ATTRIBUTE MAN'T	GET ATTRIBUTES	SUBJECT AND OBJECT ATTRIBUTES
AUTHORIZATION	AUTHORIZE	SUBJECT AND OBJECT ATTRIBUTES, CONTEXT DATA, ETC
INTERDOMAIN	FORM ASSOCIATION	SUBJECT ID AND/OR ATTRIBUTES, RESOURCE ID IN REMOTE DOMAIN
SECURITY AUDIT	AUDIT	SECURITY RELEVANT EVENT DATA
SEC. RECOVERY	NONE	
CRYPTO SUPPORT	KEY MANAGEMENT	SUBJECT ID, OBJECT ID, ASSOCIATION TYPE, ETC

4.3.4 Interactions With Communication Layer Management

Where required the Association Management Facility will issue Security Service parameters to appropriate Layer Management functions.

4.3.5 Interactions With Other Applications

Association Management needs the services of a Directory to locate requested applications and resources.

Association Management may use a Time service to time-stamp its operations.

4.3.6 Facility Management

The Association Management Facility supports the following security management functions:

- setting and changing rules for selecting Security Service parameter values
- set/change/delete/ auditable event rules.

4.4 Security State Facility

4.4.1 Introduction

The security state of a security domain is defined by the current condition of the subjects and objects that belong to that domain. If a user is successfully authenticated then this condition is recorded. The same happens when a file is opened, closed or when a user "logs off".

The Security State Facility keeps a "view" of this momentary security condition.

The Security State Facility should not be confused with the Security Audit Facility: the Security State Facility keeps the current state, not a record of state changes; however all changes of the security state should be recorded in the audit trail.

In a distributed system, security state information is maintained by all distributed elements, e.g. all Security Facilities maintain their own state information. Together this state information is "the" security state of the distributed system, modeled by a singular Security State Facility that receives state information from the other Facilities.

It should be noted that in a distributed system the information obtained from Security State can not always be fully up-to-date because of time delays involved in exchanging security state information.

4.4.2 Functions Of the Security State Facility

The Security State Facility is a passive facility that serves to hold a record of the current security state. Inputs are provided by active Security Facilities such as the Association Management Facility and the Authorization Facility. The information is read by other facilities as needed.

For example, authentication will produce a mapping between the communicating entity and an authenticated identity. This mapping, once checked, must be available to other processes. It is the function of the Security State Facility to hold such information accurately, and to provide it to other Facilities on request.

4.4.3 Interactions with other Facilities

Security State does not initiate interactions.

4.4.4 Interactions with Communication Layer Management

None.

4.4.5 Use Of Other Applications

None.

4.4.6 Facility Management

The Security State Facility supports the following security management:

None.

4.5 Security Attribute Management Facility

4.5.1 Introduction

Security attributes are assigned to the subjects and objects of a security domain. In combination with authorization rules they serve to implement a given security policy.

Security attributes assigned to subjects are referred to as privilege attributes, security attributes assigned to objects are referred to as control attributes. Since the same entity may have both subject and object roles, it may have both kinds of attributes.

Subject attribute control is closely related to authentication and the two are often considered as one single facility, particularly since the privilege attributes granted to a subject might be affected by the authentication route used. There is an important distinction between the two Facilities, which is necessary when different authorities are responsible for the introduction of users and the granting of privilege attributes.

Control attributes exist at multiple levels: the level of network objects (applications and system resources) and at the level of objects owned by applications. Access to applications and resources and access to objects and functions within applications and resources may be governed by the same attributes or by different ones. These attributes may be identical in structure or they may be different. Also, different subsets of security policy may apply to the two levels of access.

Security attributes themselves are objects owned by the Security Attribute Facility. This facility is a common facility for all of a given security domain. It may exist at the level of a network or at the level of some application that performs its own internal access control. In either case, access to the attributes must be controlled.

Central versus individual control over attributes reflects one major difference between mandatory policies and discretionary policies; see also Appendix C.

In secure distributed systems security attributes are treated as protected objects possessing their own control attributes. These are needed in the maintenance, distribution and use of the security attributes themselves.

By recursing the access control "model" it can be shown that access control to objects and access control to security attributes are different views of the same operation. Recursion is terminated when an attribute controls access to itself viewed as a protected object and/or to its peer attributes.

Many security attribute types are common to a large number of kinds of system. Examples are subject-names, subject-departments, job-roles, clearances, classifications, and the access-types associated with them. The ways in which these are represented and passed around a network would benefit from a degree of standardization. Standard ways in which a Directory may be used to support attribute management need further study.

4.5.2 Functions Of the Facility

The function of the Security Attribute Facility is the secure creation, distribution, revocation, archiving and destruction of security attributes.

The user interfaces through which security administrators and other users control the Security Attribute Facility is outside the scope of this framework.

The methods of security attribute management depend on the environment and on policy.

Transfer of security attributes between end-systems requires specific data elements and protocol.

The facility interacts as follows with other facilities in performing its functions:

- it interacts with the Security Audit Facility to record all security events specific to its operations and actions,
- it may interact with the Cryptographic Support Facility e.g. for the encryption of attributes for protection during communication,
- it will be capable of responding to security recovery information from the Security Recovery Facility.

4.5.3 Interactions With other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE SECURITY ATTRIBUTE MANAGEMENT FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED/RECEIVED
SUBJ. SPONSOR	NONE	
AUTHENTICATION	NONE	
ASSOC. MAN'T	NONE	
SECURITY STATE	NONE	
AUTHORIZATION	NONE	
INTERDOMAIN	NONE	
SECURITY AUDIT	AUDIT	AUDITABLE EVENT DATA
SEC. RECOVERY	NONE	
CRYPTO SUPPORT	ENCRYPT/DECRYPT	ATTRIBUTE DATA

4.5.4 Interactions with Communications Layer Management

None.

4.5.5 Use of Other Applications

The Security Attribute Facility may use the services of a Directory Application to store the attributes against the names of the subjects and objects that belong to the security domain.

4.5.6 Facility Management

The Security Attribute Facility requires the following security management:

- set/change/delete privilege attributes
- set/change/delete control attributes
- set/change/delete auditable event rules.

4.6 Authorization Facility

4.6.1 Introduction

Authorization rulings are made on the basis of information concerning the access requested:

1) Authorization attributes associated with the subject (privilege attributes, see Clause 4.5)

   For example the subject's name(s), its role in the system and its clearance. Indeed any attribute is a candidate for this category, provided that it is associated with the subject; in particular a name of an accessible object can be an attribute associated with a subject.

2) Authorization attributes associated with the object (control attributes, see Clause 4.5)

   For example the object's name(s), its role in the system and its degree of sensitivity or required integrity. Once again any attribute is a candidate for this category, provided that it is associated with the object. In particular a name of a subject for whom access is authorized can be an attribute associated with an object.

3) The context within which the request is being made

   For example the time of day, the communications route involved, or the accesses currently being made to other objects by this and other subjects.

   Access contexts are not further considered here, and are for further study.

4) The kind of access being requested. For example read, modify, use, know-about.

   The rules of an authorization policy are applied to values from these four categories and the result is essentially either "access permitted" or "access denied". The algorithm representing the rules can be complex, involving complex combinations of multiple elements from each category.

Any practical authorization policy must be capable of responding to changes both in terms of creation of new and deletion of old subjects and objects, and in terms of the accesses granted and withdrawn over time. It is however highly desirable that no changes to the security rules should be necessary to cater for these changes, only the data on which the rules operate. Building unnecessary understanding of the kind of access permissions to be associated with any particular privilege or control attribute must be avoided. For this reason it is appropriate to associate one or more access-type permissions associated with the attribute.

Further aspects of authorization policies and their implementations are explored in Appendices B and C.

The Authorization Facility may exist at various levels in a distributed system: it may exist at the level of a network security domain or at the level of a distributed application or a the level of a specific server.

The Authorization Facility implementations will reside in many parts of a network, most often co-located with the objects they help protect. This is not, however, a fundamental requirement since authorization needs no knowledge of subjects and objects as entities. In principle it is necessary only to present the privilege attributes and control attributes for it to reach an access authorization decision. This raises the possibility of shared supportive authorization "services" for those networks with network-wide aspects to their authorization policies.

4.6.2 Functions Of the Authorization Facility

The function of the Authorization Facility is to perform access authorization decisions on the basis of the attributes passed to it.

It makes these decision using rule sets specified by the prevailing security policy. Different rule sets may be implemented in the Authorization Facility.

In performing its function the Authorization Facility may interact with other facilities:

- it interacts with the Security State Facility to obtain state information relevant to the subject and objects involved in a given authorization request,
- it interacts with the Security Audit Facility to record all security events specific to its operations and actions,
- it will be capable of responding to security recovery information from the Security Recovery Facility.

4.6.3 Interactions With other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE AUTHORIZATION FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED/RECEIVED
SUBJ. SPONSOR	NONE	
AUTHENTICATION	NONE	
ASSOC. MAN'T	NONE	
SECURITY STATE	READ	CURRENT STATE PARAMETERS
ATTRIBUTE MAN'T	READ	OBJECT & SUBJECT ATTRIBUTES
INTERDOMAIN	NONE	
SECURITY AUDIT	AUDIT	AUDITABLE EVENT DATA
SEC. RECOVERY	NONE	
CRYPTO SUPPORT	NONE	

4.6.4 Interactions with Communications Layer Management

None.

4.6.5 Use of Other Applications

None.

4.6.6 Facility Management

The Authorization Facility requires the following security management:

- definition of authorization rule sets
- activation/de-activation of rule sets
- set/change/delete auditable event rules.

4.7 Inter-Domain Facility

4.7.1 Introduction

In practical usage of distributed systems, cooperation between different security domains will be the rule rather than the exception.

Relationships between domains will be bi-lateral and reflect underlying business arrangements. The relationship may be a peer relationship as between two domains of the same kind. The relationship may be hierarchical when one of the domains is a subdomain of the other domain. See Clause 3.1.

In cooperation between domains, subject and object related information crosses the boundary between the domains. Different domains will use different expressions of security policy. The process of translating one domain's understanding of identities and attributes into an understanding of another domain is modeled by the Interdomain Facility.

This translation may be indirect in that a commonly understood attribute transfer syntax is used between the two domains independently of their internal attribute syntaxes.

Because its function is closely related to policy there is, from a logical viewpoint, one Inter-Domain Facility within a given domain for each cooperating domain.

It is important that certain objects can be protected from access by any user from other remote security domains. However, it may be necessary to allow certain users from those domains access to other objects. Even so, in case of breaches in the security of remote domains the protected objects should remain protected. Given broad rules for access to a set of objects between domains a finer grain of control may be necessary within the object's domain based on the identity and attributes of the subject. The object's domain may have to trust the subject's domain as to the subject identity; it may also make use of subject attributes supplied by the subject's domain.

It should be noted that where there is no policy that governs interchange between domains, subjects of one (originating) domain that use the services of another (destination) domain access the destination domain through a Subject Sponsor of the latter. See also Clause 4.1.7.

4.7.2 Functions Of the Inter-Domain Facility

The Inter-Domain Facility provides a mapping between the security attributes of selected subjects

and objects of its own domain and selected external domains with which a trust relationship has been established.

This mapping function provides bi-lateral access control and data flow control between two domains.

The mapping function depends on security policy, the identities of subjects and objects involved and on their attributes.

In many ways the mapping function resembles the rule sets of the Authorization Facility. The difference between the two is that whereas the output of the Authorization Facility is a Boolean value, the output of the Inter-Domain Facility is a mapped attribute set which eventually will be used as input to the Authorization Facility.

The mapping function of the Inter-Domain Facility is two-way: requests for access may be received from within the domain or they may be received from an external domain.

In addition to its identity and attribute mapping function the Inter-Domain Facility serves to establish secure channels between two domains in order to support secure information exchange. Such secure channels may be achieved by the use of encryption.

In performing its function the Interdomain Facility may interact with other facilities:

- it interacts with the Authentication Facility in processing requests for authentication from external domains,
- it interacts with the Association Management Facility in processing requests for association from external domains,
- it interacts with the Security Audit Facility to record security events specific to its operations and actions,
- it may interact with the Cryptographic Support Facility e.g. for the encryption of attributes for protection during communication,
- it will be capable of responding to security recovery information from the Security Recovery Facility.

4.7.3 Interactions With other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE INTERDOMAIN FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED/RECEIVED
SUBJ. SPONSOR	NONE	
AUTHENTICATION	AUTHENT. REQUEST	USER ID, AUTHENTICATION INFORMATION
ASSOC. MAN'T	ASSOC. REQUEST	USER ID, OBJECT ID, ETC.
SECURITY STATE	NONE	
ATTRIBUTE MAN'T	NONE	
AUTHORIZATION	NONE	
SECURITY AUDIT	AUDIT	AUDITABLE EVENT DATA
SEC. RECOVERY	NONE	
CRYPTO SUPPORT	ENCRYPT/DECRYPT	AUTHENTICATION INFORMATION, ATTRIBUTES, ETC.

4.7.4 Interactions with Communication Layer Management

None.

4.7.5 Use of Other Applications

The Inter-Domain Facility may need the services of a Directory Application to locate objects and subjects within the domain as well as locating objects and subjects in external domains.

4.7.6 Facility Management

The Inter-Domain Facility supports the following security management functions:

- set/change/delete privilege attribute translation rule
- set/change/delete control attribute translation rule
- set/change/delete auditable event rules.

4.8 Security Audit Facility

4.8.1 Introduction

The Security Audit Facility provides security administrators with information concerning the use of the security functions of the system. Auditing has several components:

- specification of events to be audited
- audit trail collection
- audit trail analysis

- audit report
- alarm report
- audit archiving

Collection of audit trail information may be adapted to administrators needs by specifying what kind of events or occurrences coming from other facilities must be recorded. Events or occurrences may be either security violations or completions of successful operations. The operations may be either attempted actions by subjects on objects or administration functions (subject creations or deletions, key changes,...).

The collection of the events will only be possible if other security facilities send such information to the Audit Facility. To do so, these other facilities may themselves require from other facilities an explicit acknowledge or non-acknowledge at the end of specific operations.

Generally the user should not be aware of what kind of events or occurrences are or are not be recorded. However a user owning an object or having an object under his control should be allowed to specifically request the auditing of events related to that object.

Access to the recorded audit information must be limited to authorized users. Generally these will be security administrators but under some policies users may be allowed access to selected audit records concerning their own objects.

Analysis of audit trail information may occur anywhere in a spectrum that ranges from real-time analysis of events being recorded to delayed analysis of audit trails collected over a period of time. Both approaches have their specific application.

4.8.2 Functions Of The Security Audit Facility

The functions of the Security Audit Facility are:

- to provide mechanisms for the specification for events to be captured,
- to buffer the captured information in a secure manner (the audit trail information becomes a protected object owned by the Security Audit Facility)
- to analyze the audit information in search of security relevant events. This analysis may result in security alarms and in security reports.

Where this search is required to occur in real-time, the search rules and criteria must be distributed to where the audit trail information becomes first available.

- to record as audit information, all events that change its operations or that affect other facilities, e.g. the Security Recovery Facility.

These functions may be exercised remotely, so there is a need to:

- distribute specifications for auditable events,
- distribute rules and criteria for real-time security alarm generation
- distribute security alarms
- collect audit trail information
- distribute security audit reports

The information collected for the audit trail may be recorded in several locations. However, an audit report must be created in a specific location for a system administrator and therefore several audit reports coming from distributed audit trails may have to gathered to build a centralized audit report.

Audit archiving may be performed locally or centrally.

In performing its function the facility may interact with other facilities:

- it will issue event alarms to the Security Recovery Facility
- it may interact with the Cryptographic Support Facility e.g. for the encryption of audit information protection during storage and communication,
- it may invoke itself to audit some of its internal operations.

4.8.3 Interactions With other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE SECURITY AUDIT FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED
SUBJ. SPONSOR	NONE	
AUTHENTICATION	NONE	
ASSOC. MAN'T	NONE	
SECURITY STATE	NONE	
ATTRIBUTE MAN'T	NONE	
AUTHORIZATION	NONE	
INTERDOMAIN	NONE	
SECURITY AUDIT	AUDIT	AUDITABLE EVENT DATA
SEC. RECOVERY	EVENT ALARM	EVENT TYPE, EVENT SOURCE, ETC
CRYPTO SUPPORT	ENCRYPT	AUDIT TRAIL DATA

4.8.4 Interactions with Communications Layer Management

As required to audit security mechanisms in specific layers.

4.8.5 Use of Other Applications

None.

4.8.6 Facility Management

The Security Audit Facility supports the following security management functions:

- set/change/delete interchange format for audit information
- set/change/delete rules for audit information analysis

- set/change/delete specifications of alarm generating events.

4.9 Security Recovery Facility

4.9.1 Introduction

This facility is available to a security administrator to implement, in case of a suspected or real breach of security, corrective actions involving other facilities. It illustrates the clear distinction between event detection (done by auditing) and event handling.

These actions may result from a specific demand from the system administrator himself, or may be the result of events coming from the Security Audit Facility as alarms or detected security violations.

Corrective actions are specified by security policy.

The correction actions of the Recovery Facility have temporarily effects. Long term changes, needed after a breach of security, require the use of normal security management functions provided with each security facility.

Corrective actions have immediate effect and will be aimed at preventing further damage in case of a suspected breach of security. Examples are operations like immediate invalidation of passwords, immediate invalidation of all the functions performed by a user on an application, etc. In addition, the policy may require temporary modification of access control rules until such time that the system may return to its normal mode of operations.

4.9.2 Functions Of the Facility

This facility acts on events according to a set of rules defined by the security administrator. Changes or modifications to these rules should not cause any interruption of normal security management operations.

A recording of the rules must be available for the security administrator at any time. When a rule is established, changed or deleted, this action must be recorded by the Security Audit Facility.

In addition, the Recovery Facility may provide the means for a security administrator to take corrective action independently of the automatic recovery process based on the event detection capability of Security Audit Facility.

4.9.3 Interactions With other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE SECURITY RECOVERY FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED
SUBJ. SPONSOR	RECOVERY	PARAMETERS
AUTHENTICATION	RECOVERY	E.G. SUSPEND ALL/SOME CREDENTIALS FOR SUBJECT ID X
ASSOC. MAN'T	RECOVERY	E.G. ASSOCIATION TO BE TERMINATED
SECURITY STATE	NONE	
ATTRIBUTE MAN'T	RECOVERY	E.G. SUSPEND ATTRIBUTES OF X
AUTHORIZATION	RECOVERY	E.G. ACTIVATE RULE SET X
INTERDOMAIN F.	RECOVERY	E.G. ACTIVATE RULE SET Y
SECURITY AUDIT	AUDIT	AUDITABLE EVENT DATA
CRYPTO SUPPORT	RECOVERY	E.G. DISABLE KEY SET Z

4.9.4 Interactions with Communications Layer Management

The Security Recovery Facility may interact directly with Layer Management to enforce changes in associations, session- and transport connections and to change/delete crypto-variables in use at a given layer.

4.9.5 Use of Other Applications

None.

4.9.6 Facility Management

The Security Recovery Facility requires the following management:

- set/change/delete rules for taking recovery action
- set/change/delete parameters for corrective actions
- set/change/delete auditable event rules

4.10 Cryptographic Support Facility

4.10.1 Introduction

The Cryptographic Support Facility provides other Facilities and applications as well as Layer Management processes with the cryptographic functions needed for their operations.

The Cryptographic Support Facility provides support for other Facilities; it does not invoke functions of other facilities except the Security Audit Facility e.g. to audit by changes.

Variables needed for its operation are in principle provided by its "users".

4.10.2 Functions Of The Cryptographic Support Facility

The Cryptographic Support Facility provides users with the following functions:

- encryption and decryption of data e.g. records, messages, files etc.

  Users specify the kind of algorithm to be used,the mode of operation and the crypto key to be used e.g. by reference. Where data expansion occurs a specific format will be used.

- data integrity check computation.

  Users specify the kind of algorithm to be used, the mode of operation and the crypto key to be used e.g. by reference. Where data expansion occurs a specific format will be used.

- data origin authentication

  In this process the identity of the requestor is sealed with the data offered by the requestor by means of data integrity check value.

- non-repudiation of origin

  In this process the recipient of data is provided with proof of the origin of the data. This proof may be specific to a subject or to another entity such as a security domain or subdomain. The proof protects the recipient against false denial by the sender of sending the data so protected.

- non-repudiation of receipt

  In this process the sender of the data is provided with proof of delivery which will provide protection against false denial by the recipient of receiving the data so protected.

- key management

  Key management is an integral part of all operations and services that use cryptography for one purpose or another. This facility provides the key distribution and translation functions that underly key management in general. Decisions such as when to change keys between two communicating entities or the decision to replace master keys are outside the scope of this facility.

4.10.3 Interactions With other Facilities

The table below shows the interaction with other facilities initiated by this facility. The functions invoked and the information passed are given as examples.

INTERACTIONS INITIATED BY THE CRYPTOGRAPHIC SUPPORT FACILITY		
FACILITY NAME	FUNCTION INVOKED	INFORMATION PASSED
SUBJ. SPONSOR	NONE	
AUTHENTICATION	NONE	
ASSOC. MAN'T	NONE	
SECURITY STATE	NONE	
ATTRIBUTE MAN'T	NONE	
AUTHORIZATION	NONE	
INTERDOMAIN	NONE	
SECURITY AUDIT	AUDIT	AUDITABLE EVENT DATA
SEC. RECOVERY	NONE	

4.10.4 Interactions with Communications Layer Management

The Cryptographic Support Facility may be used by Layer Management processes and by Layer Entities to support their cryptography requirements.

4.10.5 Use of Other Applications

None.

4.10.6 Facility Management

The Cryptographic Support Facility requires the following management:

- secure installation of algorithms and master keys. This is provided by Security Configuration Management. See Clause 7.2.
- deletion of inactive keys
- deletion of active keys in recovery situations.
- set/change/delete auditable event rules.

4.11 Facility Interaction Matrix

The following matrix shows the interactions between the security facilities. The matrix shows initiating facilities on the left.

RESP. INVOKER	SS	AF	AMF	SSF	SAM	AUF	IDF	SAF	SRF	CSF
SUBJECT SPONSOR		I	I	I				I		I
AUTHENT. FACILITY				I			I	I		I
ASSOCIAT. MANT. FAC.				I	I	I	I	I		I
SECURITY STATE FAC.										
SEC. ATTR. MAN'T								I		I
AUTHORIS'N FACILITY				I	I		I	I		
INTERDO-MAIN FAC.		I	I					I		I
SECURITY AUDIT								I 1)	I	I
SECURITY RECOVERY	I	I	I		I	I	I	I		I
CRYPTO SUPPORT								I		

1) audit may invoke itself to audit some of its internal operation.

5. RELATIONSHIP TO THE OSI REFERENCE MODEL

Clause 4 developed the building blocks for security functions needed in distributed open systems. These building blocks can be used in two ways:

- as elements of "Supportive Security Applications" (SSA). This is a specific type of application that provides security services or security management capabilities at application level rather than embedded in the communications architecture. Note that these SSAs may be used to provide the management for security mechanisms embedded in layer processes of the OSI model. This is further discussed in Clause 6.4.3.
- as Application Service Elements or parts of Application Service Elements. An example is an ASE that handles an authorization protocol that is common across a number of applications.

5.1 Security Facilities and Application Service Elements

The Security Facilities described in Clause 4 model security functions without regard to distribution.

In secure distributed systems, security information has to be exchanged between Application Processes. Design choices and policy constraints determine the functionality of the Application Processes that perform security functions. More than one Security Facility may be implemented as part of a given Application Process. Protocol Elements common to several Application processes are modeled by Application Service Elements. There is not a one-to-one mapping between Security Facilities and Security Related ASEs. A single ASE may embody parts of different Security Facilities; conversely a single Security Facility may need more than one ASE to model its communications behaviour. See Figure 7.

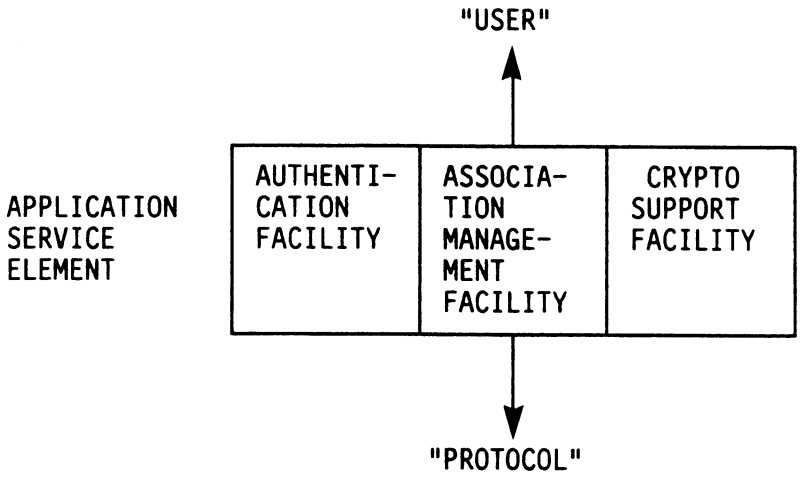


Figure 7 - Security Facilities in an Application Service Element

The various kinds of ASE related to security could be named so as to distinguish them from other types of ASES.

For example, an ASE related to secure communications could be named "Secure Communications Service Element" or SCSE.

5.2 Single Associates Objects

Single Association Objects (SAO) model the protocol generating functions specific for an association between two Application Processes. Where several SAOs use common security protocol elements these common elements may be grouped in one or more specific ASEs. The SAOs contain the Association Control Service Element, the specific ASEs and, possibly, other ASEs.

Examples of such specific ASEs are the Authentication ASE, the Confidentiality ASE and the Integrity ASE. See Figure 8.

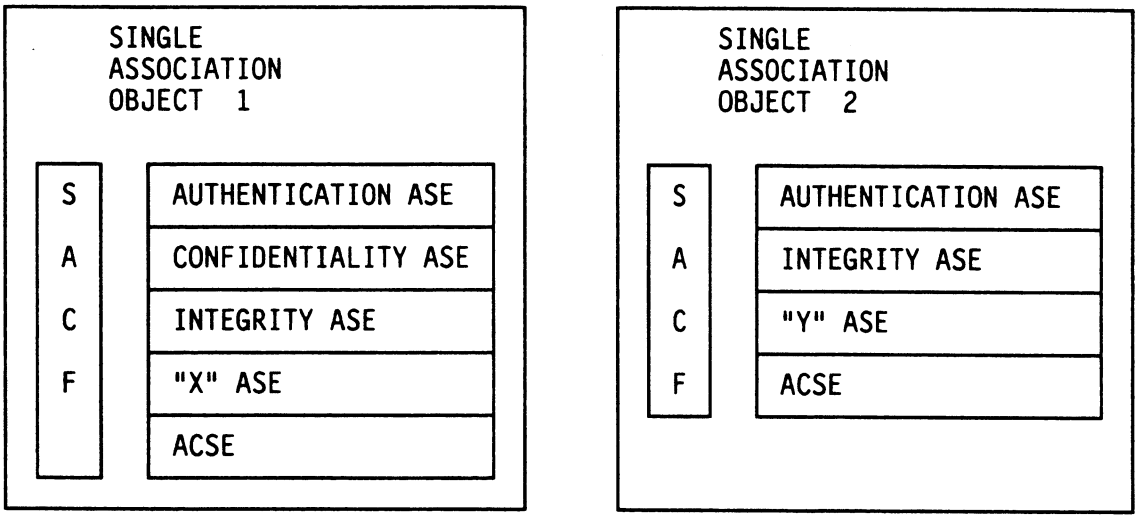


Figure 8 - Two SAOs with some common ASEs

5.3 Security Application Entity Types

Where the functionality of an application process requires independent communications behaviour towards multiple other applications the Application Entity (AE) is used as a model. The AE contains one or more SAO's. An example would be a distributed access control application that provides both authentication services and key management services.

Each AE contains SAO's and ASE's as needed for its service. Coordination between the SAO's is provided by the Multiple Association Application Service Element (MACF). See Figure 9.

For each invocation of a service the corresponding AE is invoked, this is the Application Entity Invocation.

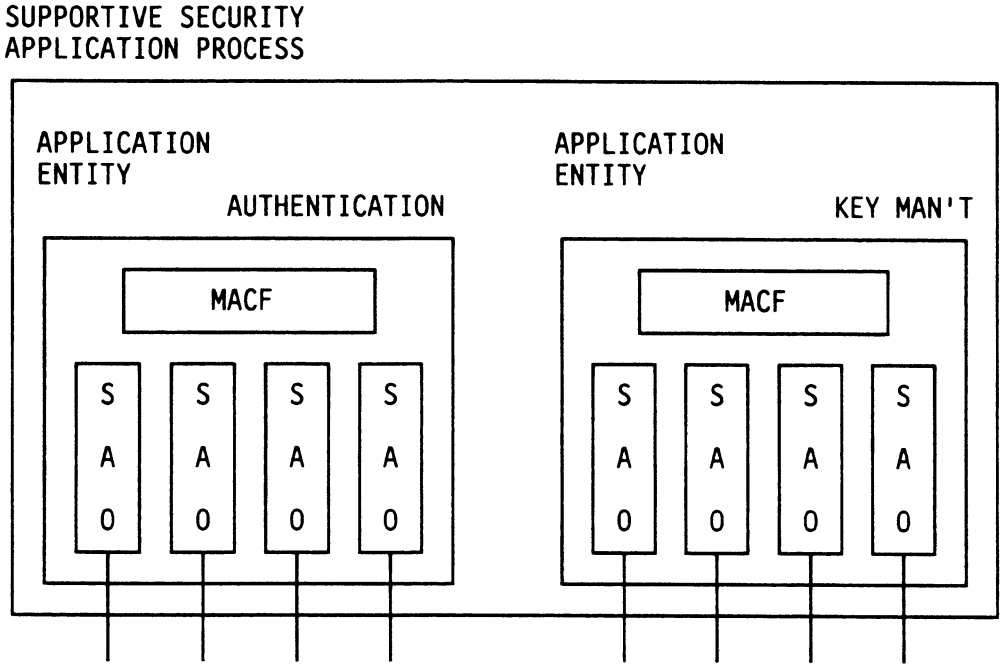


Figure 9 - Different Security Application Entities for Different Functions

6. SUPPORTIVE SECURITY APPLICATIONS

6.1 Role in The Distributed Environment

Supportive Security Applications (SSAs) provide or manage security functionality within a given security domain. In addition, an SSA may be used to provide support for the secure interworking between domains.

A set of SSAs, each with specific functionality, may support all subjects and all objects of a domain including its security administrators. The composition of this set and the composition of each SSA in terms security facilities depend on both policy and system constraints.

6.2 Client and Servers

A given SSA, irrespective of its functionality, consists of Clients and Servers as described in ECMA TR/42. The client is the part of the SSA co-located with the user application. The servers together provide the basic functionality; the clients provide users and applications with a location independent interface to the SSA.

It should be noted that from a security point of view, the servers of an SSA are necessarily "trusted functionality". Clients must always be trusted to faithfully transmit the requests of the subject, to use the proper access protocol and to do nothing not specified for their operation. Therefore, they must be subject to the proper security management.

6.2.1 Client/Server Interaction Within a Supportive Security Application

SSA Servers may be distributed throughout a distributed system. These Servers are accessed through clients. Clients use an Access Protocol to communicate with the Server. Access Protocols are asymmetrical and use Association Class 1 of ISO 9072/2; the client always initiates an interchange. Termination may come from either Client or Server. See Figure 10.

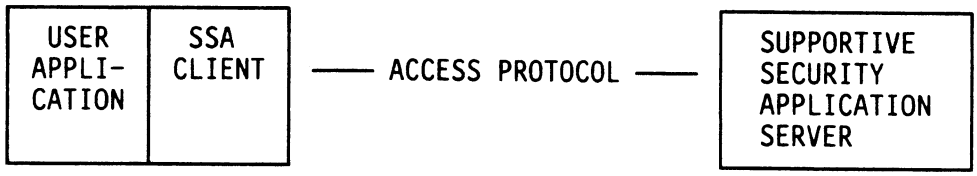


Figure 10 - Simple Model of Client and Server Components of a Supportive Security Application

6.2.2 Server/Server Interaction within a Supportive Security Application

The Supportive Security Application may be distributed application of which elements (servers) reside on different end-systems. These SSA Servers together provide a distributed security service to users located throughout the distributed system.

Communications between Servers make use of a System Protocol that is distinct from the Access Protocol used by the Clients. See Figure 11.

System Protocols are symmetrical; they use Association Class 4 of ISO 9702/2. Either Server involved in a two-way interchange may initiate the operation.

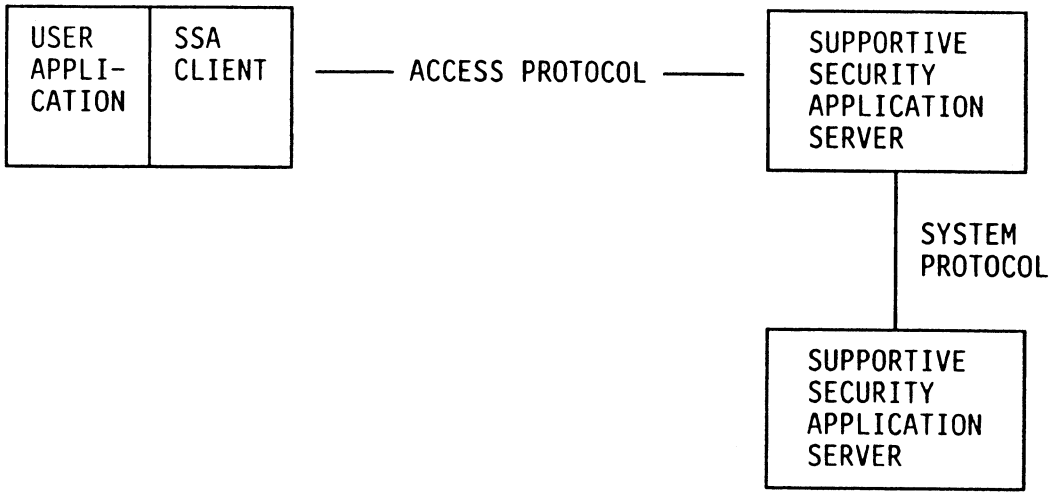


Figure 11 - Simple Model of a Distributed Security Application with two Servers and one Client

6.3 Supportive Security Applications and the OSI Reference Model

In terms of the OSI model, the level of view addressed here is at application level. The SSA's are users of the communications services defined by the Reference Model. See Figure 12.

Security Applications may use security mechanisms provided at lower layers of the Reference Model.

Under certain policies SSA's may be used as management processes that manage and control the operation of security mechanisms at lower layer of the the Reference Model. Again the interaction between Layer Management processes and the SSA's is a local concern that is outside the scope of this document.

Note that Systems Management applications are not shown in this figure so as to keep it simple.

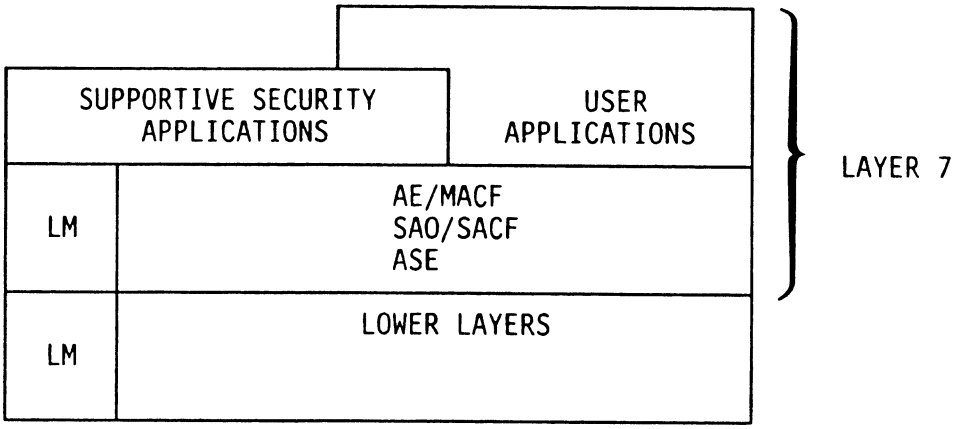


Figure 12 - Supportive Security Applications and the OSI model

1. The Security Applications interact with their peers in other systems and are users of the OSI communications facilities.
2. The lower layers may be users of the same mechanisms as the Security Applications e.g. for encryption: the Cryptographic Support Facility may be used by Layer processes.

3. Where security mechanisms are implemented at lower layers, Security Applications may be used to manage these mechanisms, e.g. through Layer Management services.

6.4 Supportive Security Application Process Structure

Figure 13 shows the result of using the OSI Upper Layer Architecture concepts to a simple model of a security server with two client applications. The server provides both authentication services and key management services.

The clients need to communicate in a secure manner but they do not share master keys with which to exchange data keys. Instead they use the security server to provide them with a shared key in a secure fashion. The clients are assumed to trust the security server and are assumed to share a master key with the server. Once a client application has authenticated itself to the security server it may request secure access to the other client application. The server will provide a copy of a shared data key for the requestor as well as for the other client.

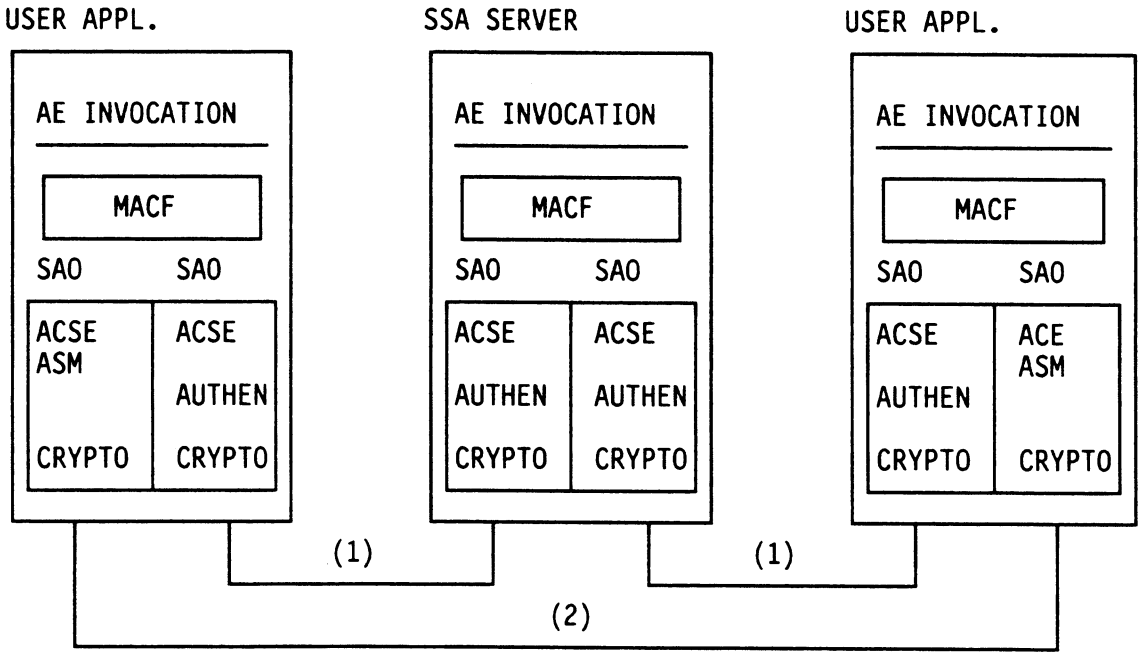


Figure 13 - Key Distribution Server into Client Applications

6.5 Service and Management Aspects

Each SSA has two distinct subsets of functionality: the "security services" it provides (e.g. authentication) and its management (e.g. credentials distribution). Thus, each SSA server process (AP) Type contains at least two Application Entity Types: an Application Entity Type = Service and an Application Entity Type = Management.

These correspond to the different protocol sets needed for use and management of a given Security Facility. In some cases these may be further subdivided into Client Types and inter-server Types of Application Entity. The former model the user and management access protocols, the latter model the system protocols needed for distributed operation.

The Security Management Application Entity (SMAE) defined in the OSI Reference Model maps to the Management Entity Type identified above.

7. SECURITY MANAGEMENT

Security Management has two major components: Operational Security Management and Security Configuration Management.

Operational Security Management is concerned with the monitoring and control of "security" in operation.

Security Configuration Management is concerned with the installation, activation and removal of security functions in a distributed system.

7.1 Operational Security Management

7.1.1 Security Management Functions

In the Framework, security functionality is modelled by means of Security Facilities. These Facilities may be used in various ways e.g. as building blocks of specialized Supportive Security Applications or as components of other processes such as user applications or management applications.

Security Management can be defined only in terms of management processes and of Security Facilities. The table below and on the following page gives an overview of the security management functions required and the facilities to which they are applicable.

SECURITY MANAGEMENT FUNCTION	APPLICABLE FACILITY
setting/changing of policy parameters e.g. timer values, actions to be taken, etc.	Subject Sponsor Facility
setting and changing rules for selecting security parameter values	Association Management Facility
create (=enter) credentials for ID x, activate/suspend credentials for ID x, delete credentials of ID x, change credentials for ID x, set/change/delete credentials change date suspend credentials for ID x	Authentication Facility
set/change/delete attributes activate/suspend attributes	Security Attribute Facility
definition of authorization rules sets suspend/activate rule sets	Authorization Facility

SECURITY MANAGEMENT FUNCTION (Continued)	APPLICABLE FACILITY (Continued)
set/change/delete privilege attribute translation rules activate/suspend interdomain operations set/change/delete control attribute translation rules	Interdomain Facility
set/change/delete interchange format for audit information set/change/delete rules for audit information analysis set/change/delete specifications of alarm generating events activate/suspend security event notification	Security Audit Facility
set/change/delete rules for taking recovery action activate/suspend corrective actions set/change/delete parameters for corrective actions	Security Recovery Facility
secure installation of algorithms activate/suspend algorithms secure installation of keys delete of active keys in recovery situations activate/suspend keys delete of inactive keys archiving of keys	Cryptographic Support Facility
set/change/delete auditable event rules activate/suspend audit data generation display current management parameters	Subject Sponsor Facility Authentication Facility Association Management Facility Security Attributes Facility Authorization Facility Inter-domain Facility Security Audit Facility Security Recovery Facility Cryptographic Support Facility

It should be noted that there is some overlap between these management functions and the security management functions defined by the OSI architecture. The extent of this duplication and its resolution need further study.

The data elements and transfer needed to implement the above management functions require further study.

7.1.2 Security Management Structures

For each Security Facility, management functionality is defined (see Clause 7.1.1). If a set of Security Facilities is described as a Security Process (e.g. a Supportive Security Application or another process incorporating one or more Security Facilities) then the management functions of the set can be described as a Management Entity that is part of the Security Process.

The concept of the Security Management Process is useful to describe the entity through which Security Administrators exercise their function.

The generic structure of Operational Security Management is represented in Figures 14 and 15. A Security Management Application Process manages the security services provided by Supportive Security Applications and by security ASE's in other applications. In the terminology of the OSI Management Framework the Security Management Application Process is a Manager and the Security Management Application Entity is an Agent. These Managers and Agents use the Management Information Transfer Services.

The use of the latter for security management purpose needs further study in particular with regard to the protection of security management information during transfer.

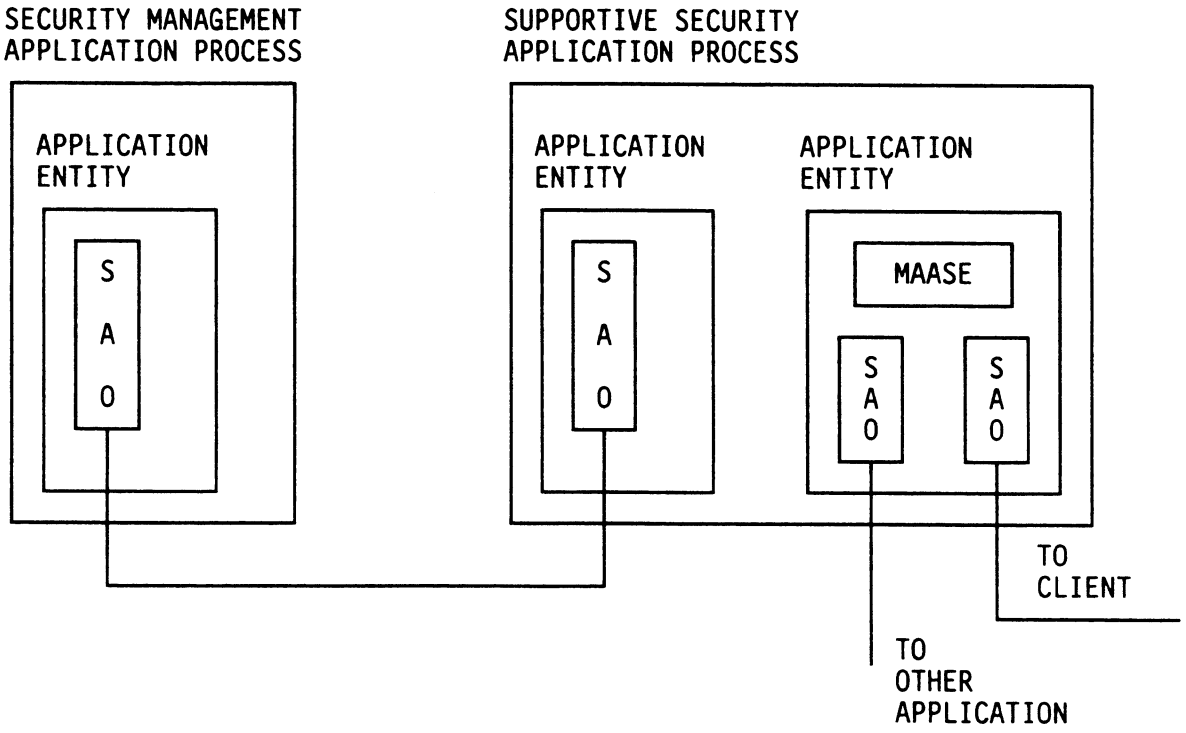


Figure 14 - Management of a Supportive Security Application Server

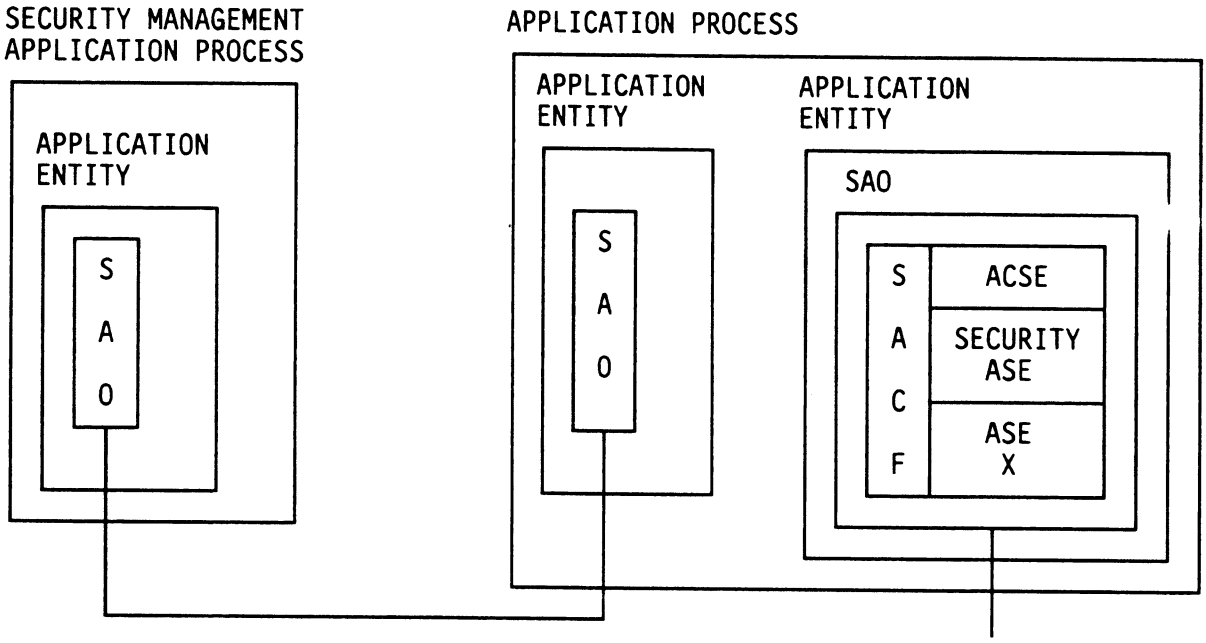


Figure 15 - Management of security function embedded in an Application Process

7.1.3 Consistency and Synchronization of Security Management

Inconsistency in the operations performed by security services may lead to weakness that could be exploited for fraudulent purposes. Also, inconsistency may give rise to unpredictable system behaviour that may have adverse effects on users, administrators and owners.

The probability of inconsistency between a pair of peer systems considered in isolation, is minimal. However, in practice distributed systems will consist of large numbers of open systems and thus inconsistency in a real concern.

Security Management, like other management, must maintain the consistency of the process it manages. This consistency must be maintained topologically - i.e. all end-systems must be controlled in the same manner - and this consistency must be contained time-wise. This is also known as synchronization.

Security Management protocols should be so designed as to assure the consistency of the security functions it manages over a wide range of conditions.

7.2 Security Configuration Management

There are parallels between Configuration Management and Security Configuration Management. The former is concerned with the installation, activation and removal of application processes; the latter is concerned with the installation, activation, de-activation and removal of Supportive Security Applications and other entities that provide security functions in distributed systems. A major difference between the two is that Security Configuration Management is concerned with the propagation of trust between systems.

In the final analysis, secure systems rely on secure hardware to store a minimum but essential set of security attributes and to execute a minimum, set of specific security procedures. Security Configuration Management is intimately linked with the management of trusted hardware; in fact, manage-

ment of trusted hardware should be considered part of it. The other major part consists of the Security Facilities described in Clause 4.

Security Configuration Management is application independent; instead it is concerned with the systems that provide the execution environment for applications. Security Configuration Management is part of the system wide security policy and it should be discussed in terms of the structure of a network security domain.

7.3 Ordering of Security Management

Generally, management is a hierarchical process that distributes the responsibility for management decisions to a number of the entities that make up a distributed system. At the top of such a hierarchy is a global Management Application Process that act as a Manager for still lower levels of Management Application Processes. At the bottom of such a hierarchy are Management Entities Application Entities that are only Agents. In Figure 16, the above is applied to security management. The Security Management Applications Entity at the bottom of this hierarchy correspond to those identified in Clause 7.1.2.

The relation of an Agent to its Manager can be understood as a subdomain that inherents its policy elements by exclusive delegation or by non-exclusive delegation.

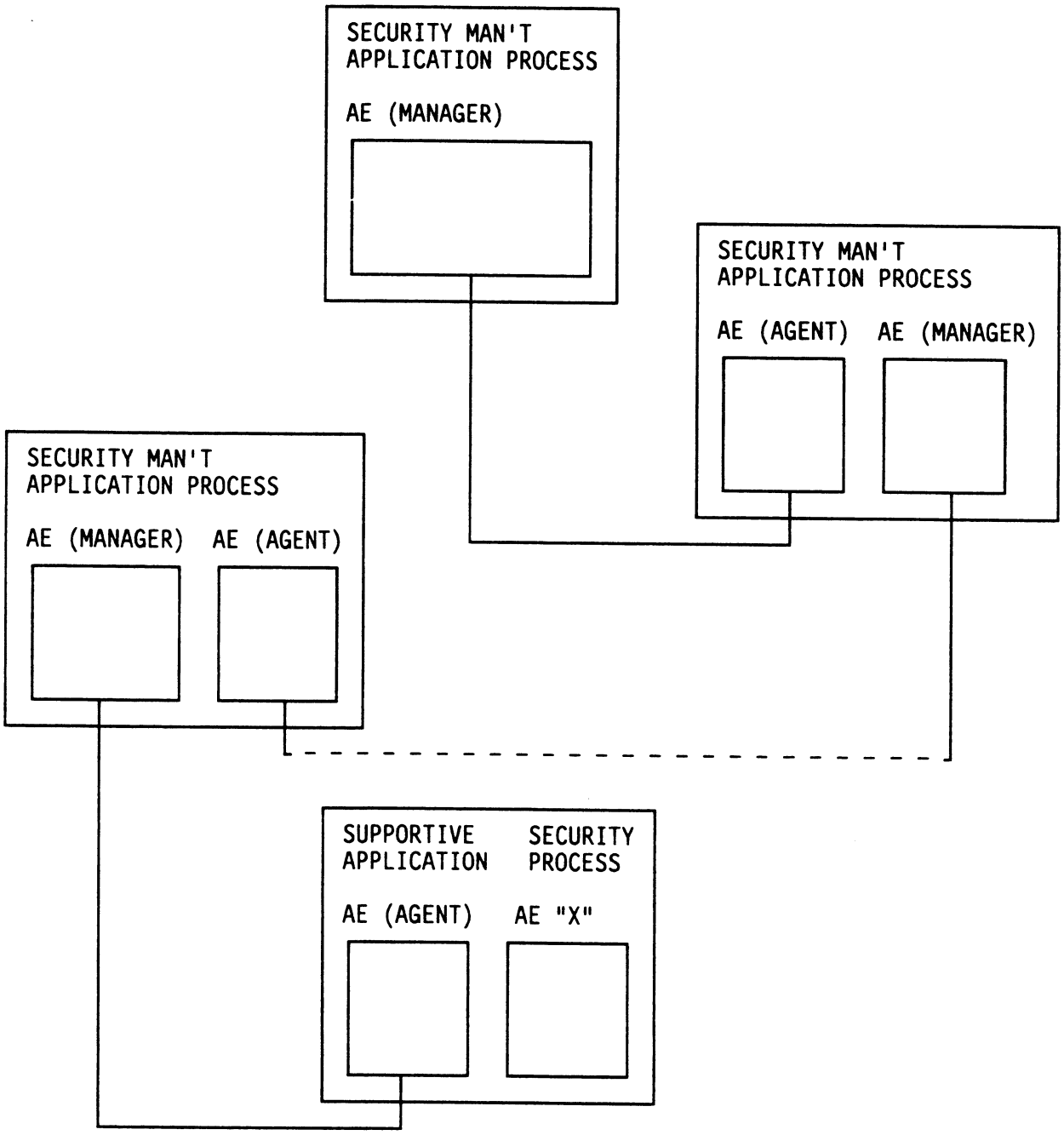


Figure 16 - Ordering of Security Management

8. CONCLUSION

This Technical Report presents a number of security requirements and provides two concepts derived from experience with secure systems by ECMA member companies and others: the Security Domain and the Security Facility.

These concepts serve as a reliable base from which Application Layer actual security services and security management tools for secure Open Distributed Systems can be developed.

The Security Framework presented here complements the OSI Reference Model and its Security Architecture. By providing a more comprehensive treatment of security and of the links between distributed security and end-systems security this Framework gives the proper perspective to other work in this field which has progressed without a similar framework.

Within ECMA this Technical Report will be the basis for further work on Data Elements, Transfer Syntax and Protocols for Security In Open Systems.

APPENDIX A

DETAILED EXAMPLE OF THE USE OF SECURITY FACILITIES IN ELECTRONIC MAIL

A.1 Introduction

This Appendix gives a detailed example of how Security Facilities interact in a specific context under a specific security policy. Although the context in this example is derived from the Mailbox Services as described in Standard ECMA-122, the sequence of interactions may equally well apply to other applications. It must be recognized that the scenario presented here shows just one way in which the Security Facilities may be used and how their functionality may be distributed in an implementation.

The scenario highlights the security aspects of establishing an association, in this case between a Mailbox Client and a Mailbox Server. They are assumed to be located in different end-systems that belong to the same security domain. For each end-system only those Security Facilities are shown that are relevant to the example.

A.2 Assumptions

The following assumptions underline the example:

1) A single security policy applies to both the Client system and the Server system;

2) this policy requires that:

   - users are authenticated before being allowed access to any application including the Mailbox Service,
   - access to the Mailbox Service is controlled at the level of the Client systems,
   - access to specific Mailboxes is controlled by the Mailbox Server systems,
   - peer-entity-authentication based on a shared crypto-key is applied at each Mailbox Server access;

3) all system functions and all Security Facilities are installed and operational;

4) peer-entity-authentication is applied during the binding of Client and Server;

5) crypto-keys needed for peer-entity-authentication are already established;

6) a Data Integrity Service at the Transport Layer is available and invocated automatically during Session establishment.

A.3 Scenario

The example given here only describes the process by which a user gets access to a specific Mailbox. It should be noted that, in addition to the security operations described here, other security controls may be applied to the functions and objects supported by a given Mailbox.

The scenario is as follows:

a) the user logs on, is authenticated and requests access to a given Mailbox identified by a Mailbox Name;

b) the Client system validates whether the user is allowed to the Mailbox Service at all;

c) if so, the Client system initiates the binding to the appropriate directory. The Client and Server system exchange peer-entity-authentication information as part of the binding process;

d) if the Client system accepts the Server's authentication information, it sends the user ID and Mailbox name to the Server for an access control check;

e) if the Server allows the access to take place the binding processes is allowed to complete. The user may now access the Mailbox, its functions and its contents;

g) once the user has finished with the Mailbox, the binding is released.

Comments:

1) the method of peer-entity-authentication described here is only an example of how this may be accomplished. Other methods are possible; this is one reason for treating peer-entity-authentication as a separate process that requires protocol elements distinct from the Associate protocol;

2) the authorization of access to a given Mailbox is in principle independent of the peer-entity-authentication. If the user would have to access another Mailbox on the same Server, a new authorization step is needed, not a new peer-entity-authentication step;

3) the user identity passed to the Mailbox Server may be distinct from the user identity used for other applications. The former may be linked to the role the user has r.e. the selected Mailbox.

A.4 Detailed Scenario

The following is a detailed, step by step description of the above scenario. The underlined step numbers refer to arrows in Figure A.1.

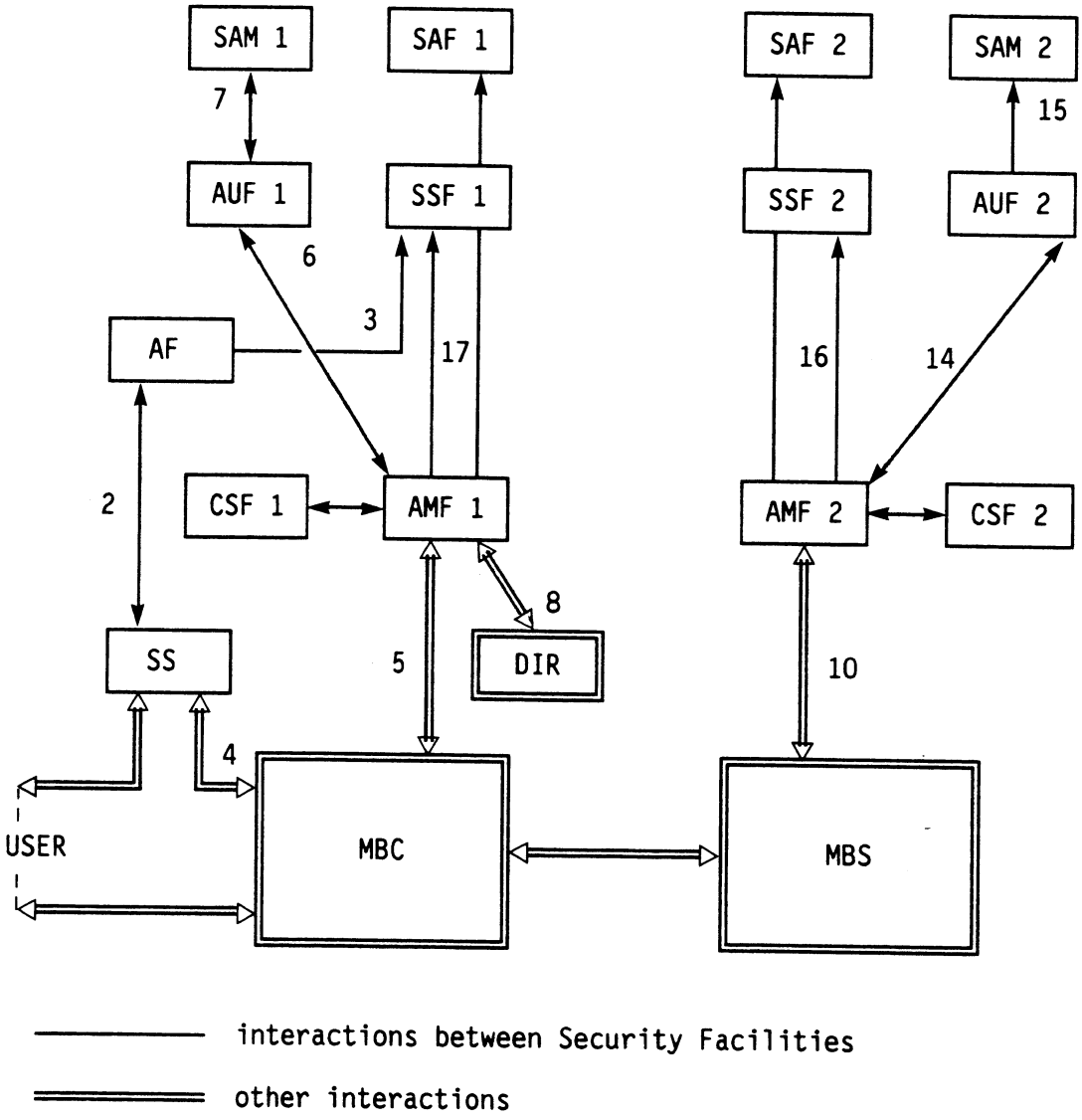


Figure A.1 - Example of Security Facilities in a Mail Service

It should be noted that the Authentication Facility is shown as co-located with the Mailbox Client. If this is not the case the essential interactions would still be the same as shown.

The abbreviations used for the Security Facilities are those used in Clause 4.11 of the main document.

1) The user activates the workstation. The Subject Sponsor (SS) is activated and requests the user's identification and authentication information (e.g. his password);

2) The SS passes the user identification and the authentication information to the Authentication Facility (AF);

3) The AF performs the authentication and the Security State (SSF1) is updated with the authentication user identity i.e. the identity under which this user is known in the security domain;

4) The SS obtains the Mailbox Name from the user and causes the invocation of the Mailbox Client (MBC) Application Entity;

5) The MBC initiates the Bind process which invokes the Association Management Facility (AMF1);

6) The AMF1 passes the user identity and Mailbox Name to the Authorization Facility (AUF1);

7) The AUF1 obtains the security attributes of the user and of the Mailbox Server from the Security Attributes Management Facility (SAM1), takes the access control decision and passes the result back to AMF1;

8) AMF1 supplies the Mailbox Name to a Directory which returns the network address (PSAP) and, possibly, other attributes, of the appropriate Mailbox Server (MBS). AMF1 returns control to the MBC;

9) The MBC invokes -Associate process. An A-Associate Request is sent to the MBS;

10) The MBS invokes AMF2 with returns a "start authentication" indication to the MBC which re-invokes AMF1;

11) The Transport Layer Data Integrity Service is activated;

12) AMF1 and AMF2 exchange peer-entity-authentication information using the Crypto Support Facility (CSF1 and CSF2) to perform the required cryptographic operations;

13) When the authentication is successful, AMF1 sends the user identity and the Mailbox Name to AMF2;

14) AMF2 invokes the Authorization Facility (AUF2) for an access control decision;

15) AUF2 invokes SAM2 to get the attributes of the Mailbox and, if applicable, the attributes of the user. It takes an access control decision;

16) AMF2 updates the local Security State Facility (SSF2) and the Audit Facility (SA2) and returns an "access granted" indication to AMF1;

17) AMF1 updates SSF1 and SA1 and allows the MBC to complete the A-Associate process;

18) The user is allowed to access the mailbox;

19) When the user is finished with the mailbox the MBC sends an A-Associate Release to the MBS;

20) The MBS invokes AMF2 indicating that the association is to be released;

21) AMF2 updates SSF2 and SA2 and returns control to the MBS;

22) The MBS returns an A-Associate Release Response to the MBC;

23) The MBC invokes AMF1 indicating that the association is released. AMF1 updates SSF1 and SA1 and returns control to the MBC;

24) The MBC returns control to the Subject Sponsor. This closes the cycle.

APPENDIX B

DISCUSSION OF SECURITY ATTRIBUTES

B.1 Introduction

There is a distinction to be drawn between the two complementary components of an access control policy: authentication and authorization. The former describes the process of providing claims of identity, the latter describes the process of controlling access by already identified subjects to already identified protected objects in a system.

This Appendix covers the topic of authorization. Its objectives are to show that existing ad hoc authorization methodologies can be fitted into a unifying framework in which apparently quite different techniques appear merely as different parts of a continuously varying spectrum. In particular the two authorization approaches characterized by capability and access control lists are shown to be extremes of this spectrum, each of which has its advantages and disadvantages.

B.2 Fundamentals

In the real world, authorization rulings are made in the context of characteristics possessed by the parties involved, the state of the world at the time, and the kind of access requested.

In the computer world we use similar concepts; they can be described in terms of the categories of input information that contribute towards determining whether a subject's request to access an object is to be granted or denied. There are four categories as follows:

B.2.1 Authorization Attributes Associated with the Subject (Privilege Attributes)

For example the subject's name(s), its role in the system and its trustworthiness. Indeed any attribute is a candidate for this category, provided that it is associated with the subject; in particular a name of an accessible object or group of objects can be an attribute associated with a subject.

B.2.2 Authorization Attributes Associated with the Object (Control Attributes)

For example the object's name(s), its role in the system and its degree of sensitivity or required integrity. Once again any attribute is a candidate for this category, provided that it is associated with the object. In particular a name of an accessing subject can be an attribute associated with an object.

B.2.3 The kind of access being requested

For example read, modify, use, know-about.

The rules of the authorization policy are applied to values from these four categories and the result is essentially either "access permitted" or "access denied" for. The algorithm representing the rules is typically complex, involving complex combinations of multiple elements from each category. One of the tasks of the standardization process is to bring some structure to this complexity in a way that preserves as far as possible its general applicability.

Notice that authorization attributes can be long lived or short lived. For example clearances and classifications tend to be static in nature, and therefore long lived. A capability on the other hand may be granted to a subject for the duration of an Association or part of an Association. In practice, short lived representations of long level attributes will be used in distributed systems for authorization decisions.

Any practical authorization policy must be capable of responding to changes both in terms of creation of new and deletion of old subjects and objects, and in terms of the accesses granted and withdrawn over time. It is however highly desirable that no changes to the security rules should be necessary to cater for these changes, only the data on which the rules operate. Thus we must not build, into the rules any unnecessary implicit understanding of the kind of access permission to be associated with any particular privilege or control attribute. For this reason it is appropriate to define an attribute as a tuple, of which one part is the attribute's value and the other is one or more access-types associated with that value.

For example, if an object has associated with it an attribute containing the name of a particular subject, paired with an access-type value of "read", there is an obvious authorization rule that could be chosen to apply under which presence of the attribute grants the subject read access to the object. Such an attribute would look remarkably like an access control list entry. A policy change which results in a requirement that this subject should have write access is achieved by means of a simple attribute value change; no rule change is required.

Not all attributes require this treatment however; for example a subject may have an attribute which defines its security clearance. Such an attribute will under many policies always be associated with read access since this is fundamental to the concept of security clearance. Such an association can therefore safely be made implicit.

B.3 Illustrative Examples

B.3.1 If we imagine an object-name/access-type tuple as a privilege attribute, with object-names also being associated with objects as control attributes, and couple these with an appropriate and obvious authorization rule we obtain what is essentially capability.

B.3.2 If we imagine a "clearance"privilege attribute and a "classification" control attribute, and couple these with an appropriate authorization rule we have a label-based scheme which is appropriate for supporting a real world National Security Policy.

B.3.3 If we imagine a subject-name/access-type tuple as a control-attribute and couple this with an appropriate authorization rule we obtain what is essentially an Access Control List entry.

B.3.4 It is easy to devise more sophisticated variants of example B.3.1 in which the object-name becomes an object type with more than one object possessing a given 'type' attribute, giving the capability a wider applicability. It is a small step further to consider this 'type' attribute as becoming a clearance (e.g. a code word) when associated with a subject, and so arrive at example B.3.2. A similar bridge could clearly be made between B.3.2 and B.3.3.

Thus clearances are revealed as generalizations of capabilities and classifications as generalizations of access control list entries.

Figure B.1 illustrates this gradual merging of one concept into another. It includes a bridge between B.3.2 and B.3.3.

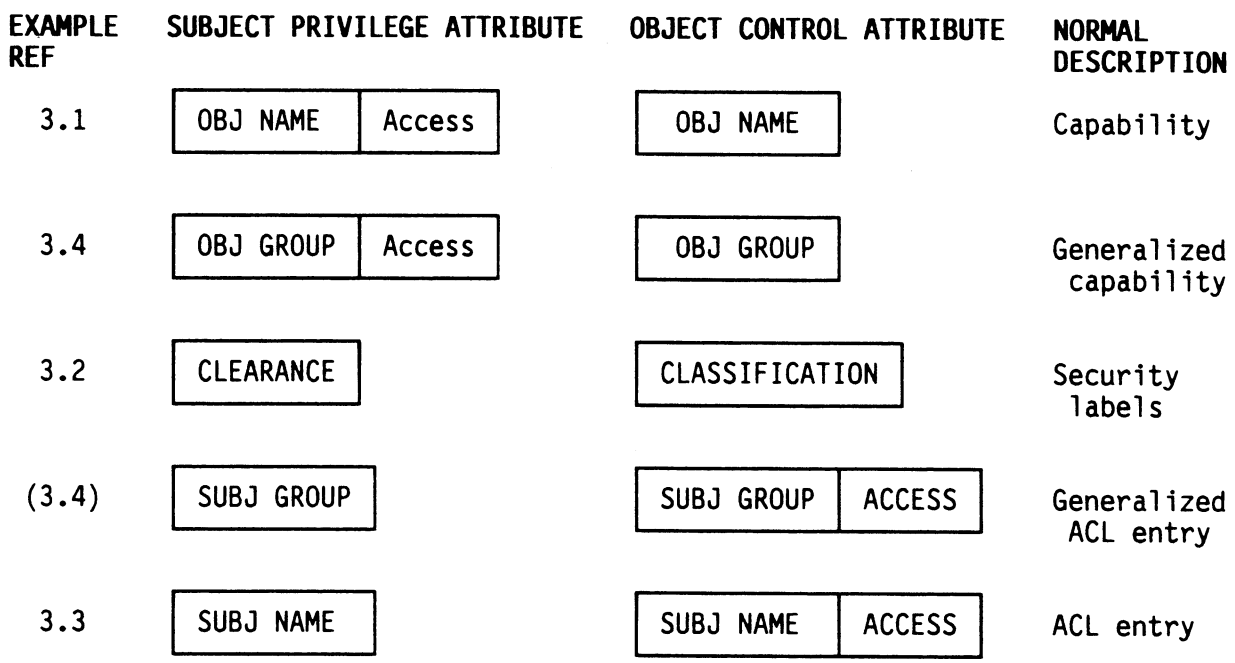


Figure B.1

B.4 Observations on the Examples

B.4.1 The ease with which a capability approach can be transmuted into a clearance/classification approach and then into a conventional access control list approach argues for the usefulness and appropriateness of the underlying attribute framework.

B.4.2 A parallel can be drawn between authorization attributes and the attributes of an entry in a directory as defined in current ISO and CCITT standards. Indeed it is quite possible that the overlap in concepts is such that directory entries held in functionally quite simple (and therefore reliable) directories may usefully be used to represent subjects with their authorization attributes and objects with theirs. These matters are for further study: at present the attributes in this paper should be viewed as abstract concepts with no particular implementation method implied.

B.4.3 Note that the access-type part of the attribute tuples proposed here do not have the same purpose as the access-type properties defined for directory attributes. The former are used in the authorization policy related to subjects and objects described by the attributes, the latter relate to the authorization policy for the attributes themselves. The latter's role is revealed by recursing the model. The attributes are now viewed as protected objects, with an authorization policy based on control-attributes which are the entries of the Directory access control lists.

This kind of recursion can easily be hidden within definitions of security policies, often because the distinction between control of access to the objects of the policy, and control over that control becomes blurred. The framework provides an approach that allows the distinction to be made clear.

Recursion is terminated when an attribute controls access to itself (viewed as a protected object) and/or to its peer attributes. Access types that help do this are:

Attribute_Read — i.e. know what the attribute contains and therefore something about the controls that are associated with the controls that are associated with the owning object.

Attribute_Write — i.e. change the control that apply to the object. Sometimes called "control".

Attribute_Copy — an access-type that would be meaningful in relation to a privilege attribute which is to be treated like a capability. It would enable the capability to be passed to another subject.

B.4.4 When subject names are held at objects for the use as control attributes (e.g. ACL entries), day to day maintenance of the authorization policy is made difficult for systems with a volatile population of subjects. Conversely, when object names are held as privilege attributes associated with subjects (e.g. as Capabilities) maintenance is difficult for systems with volatile object populations.

Maintenance is therefore clearly a factor which should influence choice of expression of policy, and to define a standard for all systems based on one or other approach is consequently inappropriate.

Furthermore, a practical system is likely to require an authorization policy which uses multiple positions on the attribute spectrum.

Figure B.2 illustrates the point.

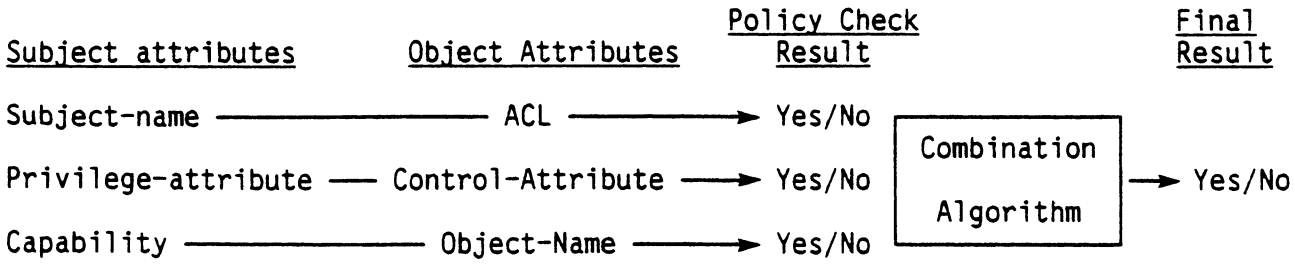


Figure B.2

Typically, high security systems might use an ACL approach for their discretionary authorization policy and a clearance/classification-attribute approach for the mandatory policy. A subject passing these tests can then (for performance reasons) be given a temporary capability which subsequently independently grants the requested access.

B.4.5 In a large distributed system, responsibility for control of an authorization policy might be devolved to a number of different centres. In particular, it will often be the case that control over the introduction of users to the network will be exercised by a different authority from those that administer the individual services, on the network. The former could be considered to be the subject administrator of the network, and the latter the object administrators. On systems with multiple cooperating authentication services there may be more than one subject administration authority.

It is useful to examine the authorization attributes that each authority controls. In general it seems obvious that subject administrators should be responsible for privilege attributes and object administrators for control attributes, with temporary attributes of either kind being allocated by object access control logic as implementation expedients. This fits in reasonably well with the real world perceptions of these attributes. It is entirely appropriate for a subject administrator to allocate user clearances, define which roles a user may assume and specify which department he or she belongs to. It is also appropriate for an object administrator to determine an object's ACL entries and security classification.

APPENDIX C

MANDATORY VERSUS DISCRETIONARY AUTHORIZATION POLICIES

A mandatory authorization policy is often thought of as a policy based on security labels, with users possessing clearances like SECRET, and protected objects possessing similarly named classifications.

A discretionary authorization policy is in contrast thought of as a policy based on individual user identity, with users being granted or denied access on the basis of who they are rather than what clearance attributes are associated with them.

Under the distributed framework these differences are revealed as merely superficial; the labels of the mandatory policy and the subject/user identity attributes associated with capability or ACL approaches are essentially the same. Indeed, if under a mandatory policy users possess unique non-hierarchic individual clearances, the clearances become equivalent to user-id's and the classifications ACL list entries.

Another distinction drawn between mandatory and discretionary policies is that mandatory policies are centrally controlled, in contrast to the discretionary policy approach based on control by ownership. In terms of the authorization framework, the difference lies in the allocation of access to the privilege and control attributes treated themselves as protected objects. Looked at in this way, it becomes apparent that a variety of choices of devolution-centralization of control is possible, depending on the authorization policy associated with the attributes. This reflects the real world requirements exemplified by security manager, sub-managers, department manager, team leaders, or individually based control policies.

A third difference drawn between mandatory and discretionary policies is that of rigour. In general, mandatory policies are expected to provide stronger protection for two main reasons:

- mandatory policies are usually implemented within an architecture which makes a clear distinction between trusted code and untrusted code. Policy control is ensured to be exercised only via trusted code.
- mandatory policies incorporate the concept of flow control (exemplified by the Bell & La Padula *property). This protects the system from malicious transfer of sensitive data to less sensitive containers by untrusted "Trojan Horse" code.

In principle however there is no reason why a discretionary policy should not incorporate such features; in practice it is operational flexibility that determines the acceptability of constraining the software contexts within which control over the policy is exercised, and it is the granularity of the authorization policy that determines the ease or difficulty of policing information flow control.