Section 5: Basic Security Services

H5

5.1 Introduction

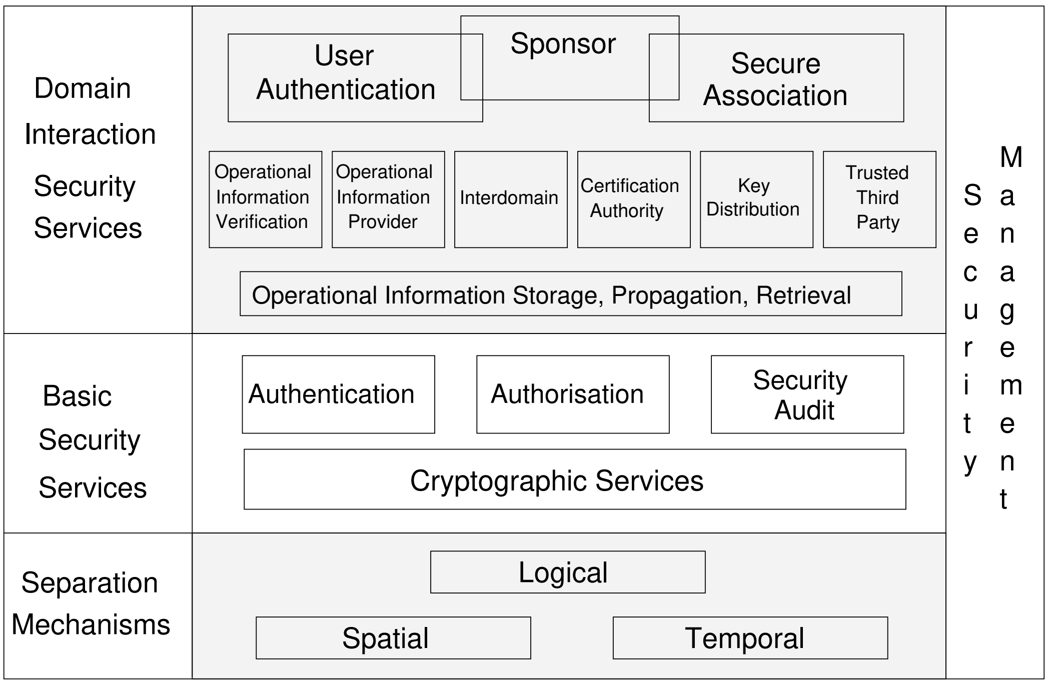


Figure 5-1 – Basic Security Services

Chapter 3 introduces the concept of layered security services. This chapter describes the services within the basic security service layer as highlighted in Figure xxx. These services are:

- cryptography and key management
- authentication
- authorisation
- security audit

5.2 Cryptographic Services

5.2.1 Introduction

Cryptography provides a basic mechanism to support higher level security services such as authentication of identities and data-origin, non-repudiation, and data separation and protection for the purposes of confidentiality and integrity.

5.2.2 Basic Model

The goal of the basic cryptographic service model, illustrated in Figure xxx, is to provide an abstraction of algorithm-specific details through a general Cryptographic Support Facility (CSF) which can sit atop different algorithms and different implementations of those algorithms (mechanisms). The CSF service interface should hide any specific mechanism, in particular any key format related to the implementation of a chosen algorithm.

The Cryptography Support Facility should be used by any application which needs to ensure the confidentiality and/or integrity of any data but is not concerned about the details of the cryptographic algorithm. Effectively this means that most applications can utilise the simplified CSF interface when invoking cryptographic services.

The CSF services include provision for data encryption, decryption, production of checkvalues (seals or signatures), and check-value-verification, and the necessary key storage and support services. Specifically, CSF provides:

- irreversible encryption
- reversible symmetric integrity and/or confidentiality protection of data
- reversible asymmetric integrity protection of data
- reversible asymmetric confidentiality protection of data
- generation of secret keys
- generation of private/public keys
- algorithm initialisation

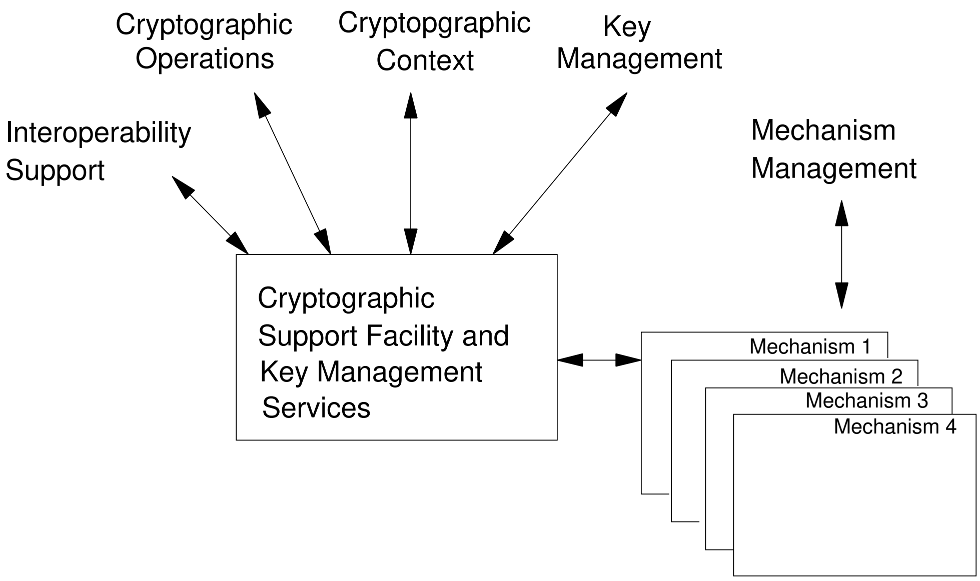


Figure 5-2 – Basic Security Services

When using the CSF, callers must call the CSF in order to create a cryptographic context. Each cryptographic context is set up for a given algorithm and key(s). The CSF returns to the caller a cryptographic context handle which is to be used in any further cryptographic operation. Once the cryptographic contexts have been established, calls to data protection interfaces are independent of the cryptographic algorithm in use and of the key storage support.

The usage model is that before using the cryptographic support facility, a client initialises a context with the required characteristics and then calls to any further data protection interface will be algorithm (mechanism) independent.

5.2.3 CSF Objects

Algorithm-Identifier

Uniquely identifies an algorithm, its mode of operation, and associated parameters (such as key or modulus length)

Algorithm-Pair

Pair of algorithm identifiers which includes at least one reversible and irreversible algorithms

Key-Information

A key value (that is, a sequence of bits) or a reference to it.

Key-Data

Data bound to a key. This data may be validity period, initialisation vector, version number, required-algorithm identifier, or usage controls (generation/verification/encryption/decryption).

Key

A key is stored by the CSF and Key-Information together with associated Key-Data.

Cryptographic-Service

The cryptographic service represents the type of transformation which is being applied to input plaintext. The cryptographic service can either specify integrity, or privacy (which is normally assumed to include integrity).

Cryptographic-Strength

The strength of cryptographic mechanism represents, in an algorithm-independent way, the effectiveness of the cryptographic transform at resisting attack. For example, the strength can be zero, weak, medium, or strong.

Cryptographic-Quality-of-Service (Cryptographic-QOS)

A Cryptographic-QOS is a Cryptographic-Service/Cryptographic-Strength pair together with an optional indication of whether symmetric or asymmetric cryptography should be used.

Cryptographic-Context

A cryptographic context encapsulates a key and algorithm and other parameters, supports cryptographic operations, and is referenced by an opaque handle.

5.2.4 Cryptographic Mechanism Management Services

Install Mechanism

This service installs an encryption mechanism for use within a system. This may include specification of the uses to which it may be applied and its assignment to support a specified Cryptographic-Quality-of-Service.

Deinstall Mechanism

This service removes a mechanism from a system.

Disable Mechanism

This service disables a mechanism and thus inhibits its use.

Re-enable Mechanism

This service re-enables a mechanism previously disabled.

5.2.5 Key Management Services

Generate-Random-Number

This function generates a random number.

Generate-Secret-Key

This function generates a key from an input key size, algorithm, and Key-Data. It returns a Key-Handle.

Generate-Asymmetric-Key-Pair

This function generates a public/private key pair according to an input algorithm identifier.

Initialise-Key

This function is used to hide a key below the CSF API. It returns a Key-Handle to the input key.

Get-Key-Information

This function returns the Key-Information associated with the input key handle. Access to this information may not be open to all callers.

Get-Key-Data

This function returns the Key-Data associated with the key handle.

Release-Key

This function releases the key associated with the input key handle. Any cryptographic context set up with that key handle is no longer accepted by the CSF, although the cryptographic context is not released by this service.

5.2.6 Public Key Management

Domain interaction security service applications (such as operational information provider services, key distribution services, or trusted third parties), or any security-aware application may need access to public key management services to:

- hold asymmetric key pairs, and
- require access to the certificates that are used to bind the identity associated with each public key.

A public key management service provider (PKM) encapsulates the mechanisms used to manage public keys and their certificates. This includes all steps from generation of an asymmetric key pair, to its final expiry or revocation.

The PKM comprises the following:

- An operational interface providing access to certificate and key handling services
- A Certification Authority (see
- An administration service used to generate, certify and revoke key pairs

The PKM services are as follows:

Get-Public-Key

Returns a public key for a specified principal, and ensure that the validity of the public key is verified (using appropriate certificates, uncertified public keys, revocation lists, etc.). The caller can optionally specify a certificate identity, then the related certificate is guaranteed to have been used when verifying the key.

Get-Private-Key

With the same inputs as Get-Public Key, this function does the same checks, but instead returns the private key.

Query-Certificates

Used to find out which certificates are involved in validating a public key. Security-aware applications can utilise this additional information to select a public key from a number of possible alternatives.

Check-Validity

Used to check the remaining validity time for a public key.

Store-Certificate

Add the supplied certificate for storage in the most global storage area accessible.

Put-Uncertified-Public-Key

Installs an uncertified public key into PKM. This results in the underlying cryptographic support facility being invoked to initialise and store the key using a mechanism appropriate to the algorithm.

Put-Uncertified-Key-Pair

Installs an uncertified public key and its corresponding private key into PKM.

Delete-Key

Removes a public/private key from PKM storage.

Get-Other-Key

Given one of the keys of an asymmetric key pair, returns the other if it is available and also accessible by the caller.

5.2.7 Operational Services

These services support the creation, modification, and deletion of cryptographic contexts, and allow callers to apply, and where appropriate, reverse cryptographic transforms on data. These services may be used by applications or by security service providers, for example authentication and access control information service providers for the generation of security certificates or security tokens.

Context Management Services

These services support the creation, modification, and deletion of cryptographic contexts.

Create Cryptographic Context

This service enables the caller to create a context within which subsequent cryptographic operations are performed using an algorithm which the caller need not be aware of. The caller may specify a quality of service or other parameters in order to control the strength and nature of the underlying algorithm

Initialise-Context

This function creates and initialises the Cryptographic-Context with the input parameters and returns a pointer to its handle. Otherwise the context is modified according to the input parameters. The input parameters comprise a CSF key handle, and an Algorithm-Pair,

Modify-Context

This function modifies an existing Cryptographic-Context with the specified input parameters.

Duplicate-Context

This function duplicates the specified Cryptographic-Context and returns a handle to the replica. It may be used in conjunction with Initialise-Context to produce a modified version of the context (and associated) key without releasing the current context.

Release-Context

The function releases the specified Cryptographic-Context, that is, it can no longer be used for any cryptographic operation. This operation does not release the key attached to the key handle contained in the released context.

5.2.8 Cryptographic Operations

These services allow callers to apply, and where appropriate, reverse cryptographic transforms on data.

One-Way-Function

This function irreversibly encrypts the input text. The input context handle refers to a Cryptographic-Context containing a One-Way-Function Algorithm-Identifier.

Encrypt

The function returns the input text encrypted using the reversible cryptographic algorithm and the key as specified in the confidentiality data protection Cryptographic-Context.

Decrypt

The function returns the input text decrypted using the arguments specified in the Cryptographic-Context.

Generate-Check-Value

The function returns the seal of the input text computed using the cryptographic algorithms and the key as specified in the integrity protection Cryptographic-Context. This service applies a signature to data prior to, or as part of data transfer for purposes of integrity or data-origin authentication.

NOTE: This service is also referred to as a message digest service.

Verify-Check-Value

This service validates the integrity of the contents of a message, and authenticates the origin of a message against the supplied Check-Value previously created by Generate-Check-Value. The function checks the integrity of the input text using the Cryptographic-Context which may include a hash algorithm. The seal of the input text is computed using the algorithm(s) specified in the Cryptographic-Context. The result is then compared with the given proof of integrity. It returns true in case of equality, false otherwise.

5.2.9 Interoperability-support Operations

Extract-Key

This function packs the key attached to the input handle into an exportable octet string format.

Restore-Key

This function unpacks the key handle contained in an octet string and returns it to the caller.

Extract-Context

This function packs the Cryptographic-Context into a string of octets.

Restore-Context

This function rebuilds a Cryptographic-Context handle using an input octet string and returns it to the caller.

5.2.10 Impact of Cryptographic Service on Primary Service APIs

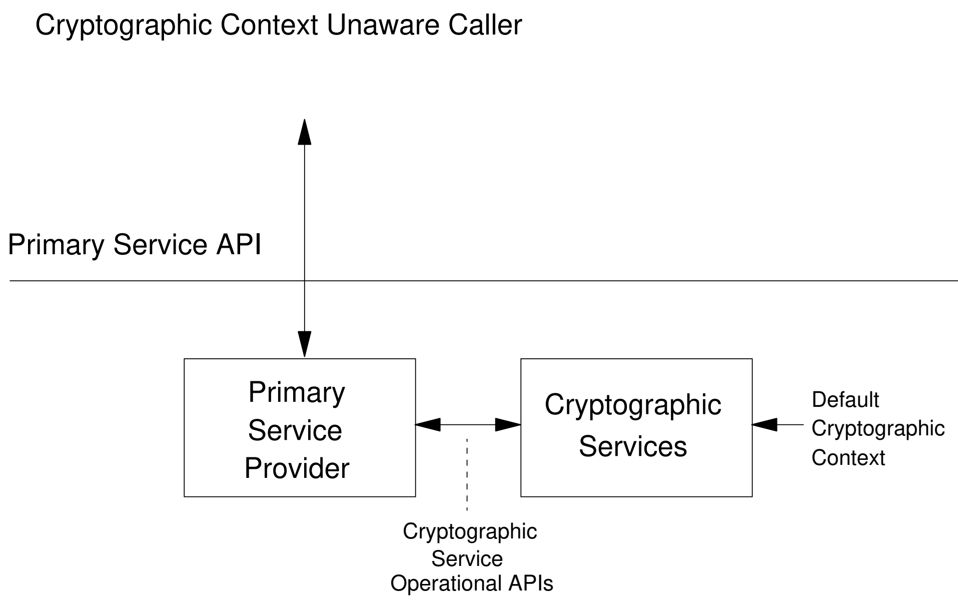


Figure 5-3 – Cryptographic Context Unaware Caller

In the case of a cryptographic unaware caller, the caller of the primary service API is unaware of, and not responsible for managing, the cryptographic context. The primary service API provides no support for the caller of the interface to specify the cryptographic context or cryptographic mechanisms to be used.

The cryptographic context is taken as a default cryptographic context previously established and associated with the tree of execution making the call.

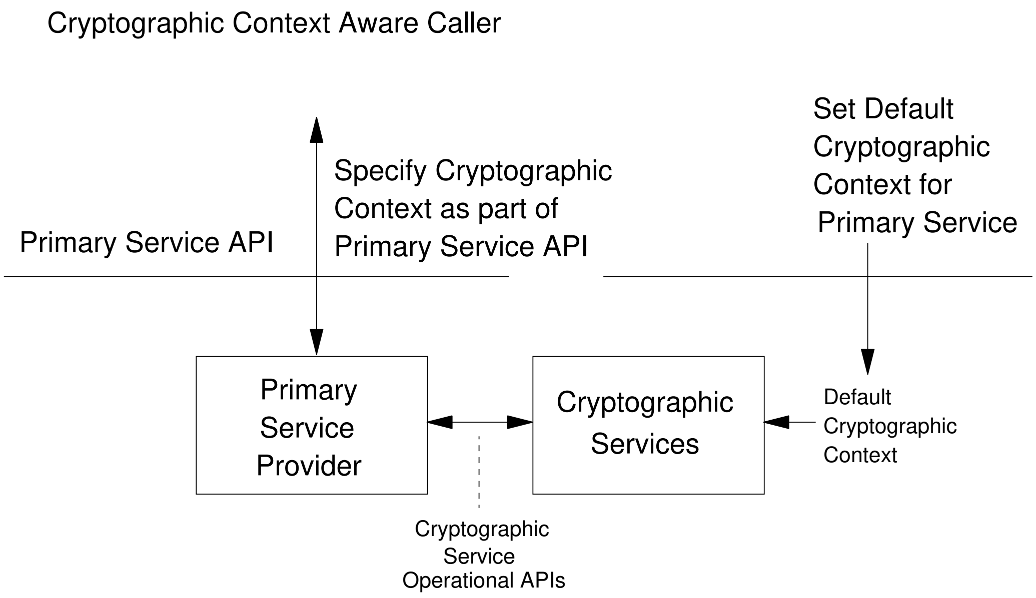


Figure 5-4 – Cryptographic Context Aware Caller

In the case of a cryptographic context aware caller, the caller of the primary service API is aware of, and responsible for managing, the cryptographic context. The primary service API may provide support for the caller of the interface to specify the cryptographic context or cryptographic mechanisms to be used as part of the primary service API. Alternatively the caller may be required to specify the cryptographic context in an out-of-band manner by setting the default security context to be applied to any cryptographic service calls arising from invocation of the primary service API.

5.3 Authentication

5.3.1 Introduction

Authentication services provide corroboration of the claimed identity of an entity. An entity's authenticated identity is used by other security services and operations on behalf of the entity. Consequently the deployment and use of strong identification and authentication services and mechanisms is fundamental to the success of the overall security solution. These services and mechanisms are required at the points of access to the information systems and the services they provide to effectively counter unauthorised use through masquerade.

The basic authentication strategy is to use either secret or unforgeable and unique information shared by the requesting entity and the identification and authentication mechanisms, possibly on both sides. Examples include passwords, secret information held on tokens or smart cards, or biometric information (for example finger prints, retinal scans) unique to each entity. The primary mechanism used to protect against unauthorised disclosure of an entity's secret credentials is cryptography.

In general there are two forms of authentication;

— entity authentication, and

— data origin authentication.

Entity authentication provides corroboration of the identity of a principal, while data origin authentication provides corroboration of the identity of the principal responsible for the creation of a specific data item.

5.3.2 Basic Authentication Service Model

Figure xxx illustrates the basic ISO model of authentication [ISO 10181-2] followed by an explanation of the components of the authentication process: /f2This needs proper use of Cross Reference macro./fP

The *claimant* is the entity representing a principal for the purposes of authentication.

The *verifier* is the entity representative of the entity requiring an authenticated identity.

The *trusted third party* is a security authority or its agent that is trusted by the other entities with respect to security-related activities.

NOTE: Is this the best term to use here? We have used Trusted Third Party under Domain Interaction Services with specific reference to Notarisation and Non-Repudiation.

Claim AI is information needed to initiate the authentication process on behalf of the principal, for example, a password, a secret key or a private key.

Verification AI is used to verify an identity claimed through exchange AI, for example *password* related to identity of principal, *secret key* related to identity of principal or authority, *public key* related to identity of principal or authority.

Exchange AI is information exchanged between a claimant and a verifier during the process of authenticating a principal. Examples are: *claimed distinguishing identifier* , password, challenge, response to challenge, test number, verifier identifier, result of transformation function, on-line certificate, off-line certificate.

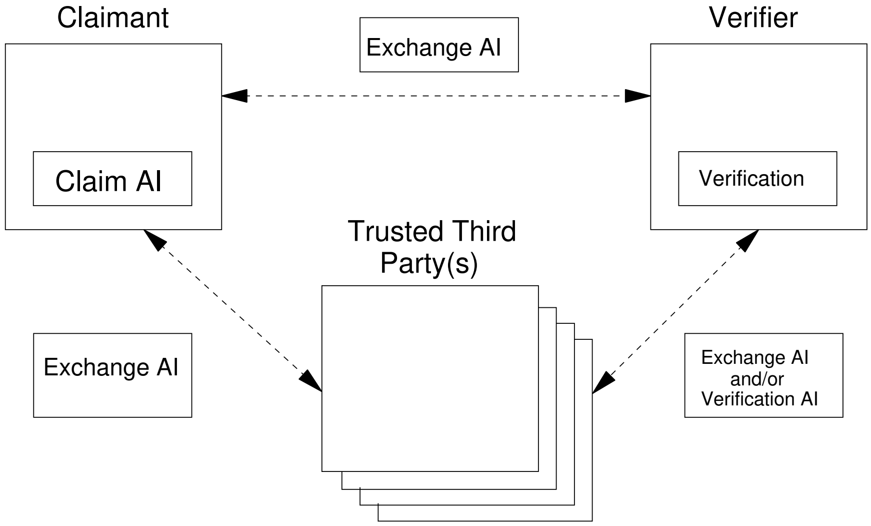


Figure 5-5 – Basic Authentication Service Model

Further explanations on the variations of this model can be found in the ISO/IEC DIS 10181-2.2 document, namely No Third Party, In-line Authentication, On-line Authentication, and Off-Line Authentication.

The generic security service model divides a security service into two categories of services: management services and operational services.

5.3.3 Management Services

Authentication management services include distribution of descriptive information, passwords or keys, to entities required to perform authentication. Authentication management services include install, de-install, change AI, distribute, disable and re-enable.

A brief explanation of each management service follows:

Install

Installs claim AI and verification AI into the authentication service SMIB. The install operation may be a compound operation comprising:

enrollment
: the recording of authentication information associated with a principal

testify
: the provision of additional information by a witness entity

validate
: introduces a principal into a security domain (activates a principal record) and is performed on behalf of a security authority. It may also be used to revalidate a principal previously invalidated

confirm
: invoked after a validate service, returns specific information such as acknowledgement or rejection of enrollment to a principal.

De-install

De-install causes a principal to be removed from a population of authenticatable principals. De-install may be refined with:

invalidate
: causes a principal to be disabled from authenticating, but leaves installation record present in SMIB

notify
: may be invoked by security authority after invalidate service to notify principal of its invalidation and possibly information on how to re-enroll.

unenroll
: causes a principal to be suppressed from a security domain.

Change AI

Change AI is invoked on behalf of a principal or security authority to cause authentication information to change, for example change password

Distribute

Distribute is a service that enables any entity to acquire sufficient verification AI upon which to verify exchange AI. An example is the distribution of cryptographic keys used to verify cryptographic seals applied to authentication certificates.

Disable and Re-enable

Disable causes a state to be established whereby a principal is temporarily unable to authenticate. Re-enable causes the disabled state to be terminated.

5.3.4 Operational Services

Authentication operational services are those services invoked as part of the actual authentication process. Authentication operational services include acquire, generate and verify.

A brief explanation of each operational service follows:

Acquire

Acquire allows a claimant or verifier to obtain information required to generate exchange AI for an instance of authentication. This may require interacting with a trusted third party (for example, authentication service).

Generate

Generate is invoked by a claimant to generate exchange AI, or to process received exchange AI.

Verify

Verify is invoked by a verifier to verify received exchange AI from the claimant, or to generate exchange AI for transfer to the claimant.

5.3.5 Impact of Authentication Service on Primary Service APIs

Within an IT System the necessity to authenticate a principal's identity arises in two specific circumstances: firstly, the identification and authentication of a user, physically distinct from the IT system, and secondly, as part of the creation of a secure association between security domains within an IT system, or between IT systems.

The authentication service is a service that supports the formation of associations between entities. As such the impact with primary services is covered under the section on Secure Association Service.

5.4 Authorisation Service

5.4.1 Introduction

Authorisation services are used to counter unauthorised modification and disclosure threats.

A range of authorisation and authentication services can be combined to form an access control regime. The general purpose is to ensure that only those entities (known as initiators) with the necessary authorisations are allowed to access information and information system resources and services (known as targets).

Authorisation services may be used at many places within a system. Within a single security domain it may be used:

- as part of the initial authentication of a principal at the granularity of access to the domain as a whole
- at the granularity of access to an activity, or group of activities, supported by the domain
- at the granularity of an element, or group of elements, of the domain
- as a combination of the above.

When a system function is provided by a combination of security domains within a system, in particular a combination of service domains and platform domains, the overall system access control policy is enforced by the combination of authentication and authorisation services within all the constituent security domains.

This section describes the ISO model of access control. This is based upon the concept of making an access authorisation decision (the authorisation service) and then enforcing that decision.

5.4.2 Basic Service Model

Figure xxx illustrates the basic model of authorisation which is described in terms of Access Control Information and Access Control Functions.

Access Control Information (ACI) is information used for access control purposes. ACI may be associated with entities such as initiators or targets, may be associated with actions, and may include contextual information.

Access Control Decision Information (ADI) is the portion (possibly all) of the ACI associated with an entity or action that is made available for use in making a particular access control decision.

The basic entities and functions involved in access control are the initiator, the access control enforcement function (AEF), the access control decision function (ADF), and the target.

The AEF is responsible for ensuring that any actions by the initiator on the target are authorised (by the ADF). When an initiator makes a request to perform an action on the target, the AEF invokes the services of the ADF so that a decision can be made. In order to make the decision, the ADF is provided with, or acquires Access Control Decision Information (ADI) associated with the initiator, the target and the action. It is the responsibility of the caller of the ADF to ensure that the ADI has been verified before it is used as input to the access decision. Other inputs to the ADF are the access control policy rules and contextual information needed to interpret the ADI or the policy. Contextual information may include location, time of access or communication path.

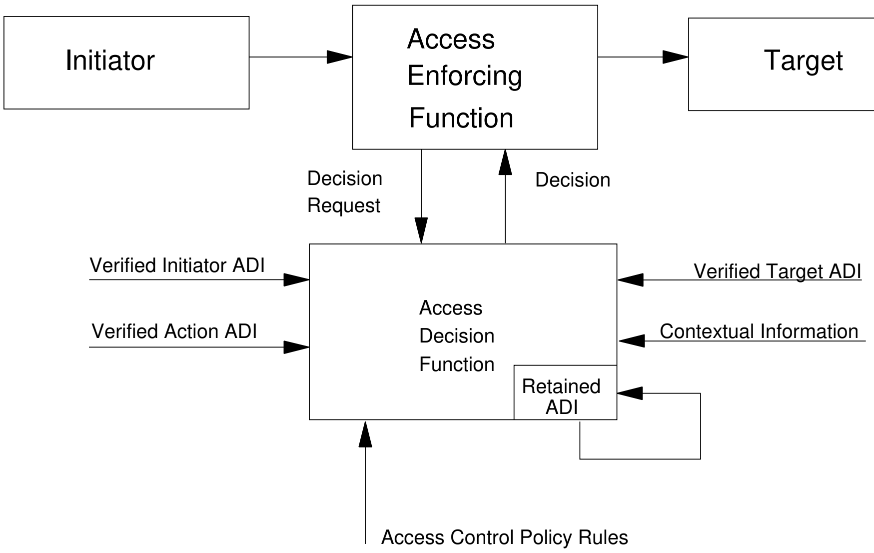


Figure 5-6 – Basic Authorisation Service Model

When the AEF requests a decision of the ADF the ADF requires access to the appropriate ADI and contextual information. This may be:

• preplaced at one or more ADF components after allocation of the ACI values

• ADI may be delivered to the ADF as part of the decision request,

• ADI may be obtained from another functional element (for example, a Directory Service Agent). The ADI may be obtained by the initiator or target, or by the ADF itself.

As described in the enterprise security authority as part of the system design procedure maps enterprise security policy rules to authorisation services supported by the system and maps enterprise security attributes to ACI within the system.

The ISO framework includes the following examples of Access Control Information:

• Initiator ACI:

— individual access control identities

— identifier of hierarchical group in which membership is asserted

— identifier of functional group in which membership is asserted

— role that may be taken

— sensitivity markings to which access is allowed

— integrity markings to which access is allowed

— a target access control identity and the actions allowed on the target — that is a *capability*

• Target ACI:

— target access control identities

— individual initiator access control identities and the actions on the target allowed or denied them

— hierarchical group membership access control identities and the actions on the target allowed or denied them

— functional group membership access control identities and the actions on the target allowed or denied them

— role access control identities and the actions on the target allowed or denied them

— authorities and the actions authorised for them

— sensitivity markings

— integrity markings

• Action ADI

— ADI associated with operands of the action (data ADI), for example:

— sensitivity markings

— integrity markings

— originator identity

— owner identity

— ADI associated with the action as a whole, for example:

— initiator ADI

— permitted initiator target pairs

— permitted targets

— permitted initiators

— allowed class of operations (for example, read, write)

— required integrity level

• Contextual Information:

— time periods

— route (an access may be granted only if the route being used has specific characteristics)

— location (an access may be granted only to initiators at specific end-systems, workstations or terminals, or only to initiators at a specific physical location).

— system status (an access may be granted only for particular ADI when the system has a particular status, for example during a disaster recovery)

— strength of authentication (an access may only be granted when authentication mechanisms of at least a given strength are used)

— other access currently active for this or other initiators

ACI has to be allocated to elements by the Security Authority. This is part of the process of installing or creating elements within the domain. ACI must be allocated to initiators before they may request any actions. ACI must be allocated to a target before it may be used. The allocation of ACI may require specific administrative action on the part of the Security Authority, for example configuration of ACI for a principal to be permitted to utilise the services of a domain, or may be performed automatically as part of the element creation process on the basis of the security context of the action creating the element, for example, creation of a file.

ACI and ADI require binding to the elements of a domain, including a sponsor and principal pairing as an initiator, to provide assurance to the access control functions that the ACI or ADI is correctly associated with the element. The ACI, and its binding to an element, may be propagated. Therefore, the integrity of the ACI and its binding to an element requires protection when the ACI or the entity are stored, processed, or exchanged. The methods by which ACI may be bound to elements include:

• on the basis of storage location, for example, storage of security attributes as part of the inode of a file

• on the basis of entity identity combined with a security attribute security service, this may require the support of an authentication service to verify an entity's identity (see User Identification and Authentication)

• on the basis of cryptographic processes for the signing or sealing of sets of information, particularly relevant in communication services (see Secure Association Service).

The functions that support the binding of ACI also need to support the selection of subsets of an element's ACI by the element and the binding of a subset of the ACI to another element supporting the delegation of access rights to that element as proxy for the initiating element.

If the ACI, or part of the ACI, associated with a principal are revoked, any ADI generated on the basis of that ACI is also required to be revoked. This may be achieved by specific action or by relying on the time expiry of such ADI when the ADI takes the form of a security certificate or security token.

5.4.3 Classification of Authorisation Schemes

Both ECMA and ISO present a classification of authorisation schemes which demonstrate that the various schemes can be considered as different parts of a continuous spectrum. This illustrates that the concepts applied are consistent. The variations occur in where the ACI are held and managed. The basic schemes may be categorized as:

- Capability Scheme

ACI comprising target-name/access-type tuples assigned to initiators provides the basis of a capability based authorisation scheme. Such a scheme is difficult to administer with a dynamic target population.

A generalised capability scheme may assign tuples comprising a target-group-name/access-type. This can provide authorisation to access types of target rather than specific targets. A specific example is the assignment of authorisations to initiators which grant authority to utilise specific functions in respect of all targets appropriate to the function. This use of capabilities within an operating system service context is often referred to as privilege. However, within this framework privilege has the wider meaning of access rights in a total sense.

- Label Scheme

ACI associated with both initiators and targets together with an authorisation rule based on comparison of the initiator and target ACI provides the basis of a label based scheme (sometimes referred to as a Mandatory Access Control scheme.)

- Access Control List Scheme

ACI comprising initiator-name/access-type tuples assigned to targets provide the basis of a access control based authorisation scheme. Such a scheme is difficult to administer with a dynamic population of initiators.

A limited form of such a scheme is the permission bit scheme supported by POSIX 1003.1/ISO 9945-1.

NOTE: *This needs proper use of Xref macros.*

The spectrum of authorisation schemes is illustrated in Figure xxx.

In practice a combination of authorisation schemes is applied with the authorisation decision depending on the outcome of the combination of the authorisation decisions under each applicable scheme. This is illustrated in Figure xxx. The combination algorithm depends on the implementation. As an example an algorithm may require access to be authorised by both a label-based scheme and an

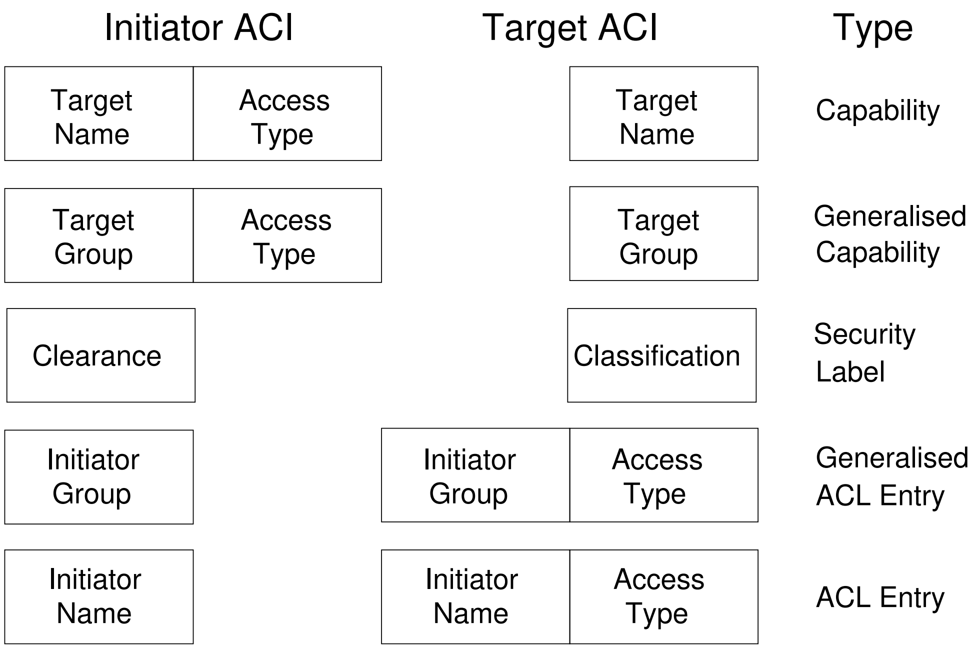


Figure 5-7 – Spectrum of Authorisation Schemes

ACL-based scheme or by a capability-based scheme.

Figure xxx is an illustration of a combination algorithm. In this example authorisation needs to be granted by both the ACL authorisation scheme and the label based scheme. Authorisation under the ACL scheme is granted by an appropriate entry in the ACL or by the initiator possessing an ACL-Capability. Authorisation under the Label scheme is granted by the comparison of the initiator and target labels and the label scheme rule or by the initiator possessing a Label-Capability.

5.4.4 Discretionary/Non-discretionary Authorisation Schemes

Within a security domain an authorisation scheme may be a discretionary authorisation scheme or a non-discretionary authorisation scheme.

A discretionary authorisation scheme is one under which any principal using the domain services may be authorised to assign or modify ACI such that he may modify the authorisations of other principals under the scheme. A typical example is an ACL scheme which is often referred to as Discretionary Access Control (DAC).

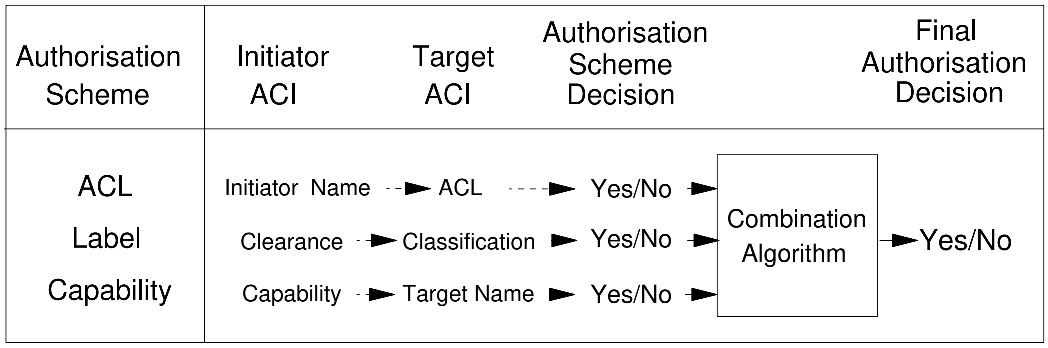


Figure 5-8 – Combination of Authorisation Schemes

A non-discretionary scheme is one under which only the recognised security authority of the security domain may assign or modify ACI for the authorisation scheme. This means that the security services within the security domain are responsible for the assignment of ACI when entities are created. A typical example is a label based authorisation scheme used as part of an information flow policy based upon information sensitivity. Such a scheme is often referred to as Mandatory Access Control (MAC).

An authorisation scheme may have both discretionary and non-discretionary aspects. Thus a principal may be authorised to assign his own authorisations or a subset thereof to other principals or to make the set of authorisations more restrictive. An example under a label-based scheme is that principals may be authorised to increase the sensitivity of information, and thus potentially reduce the set of principals then authorised to access the information, but not to decrease the sensitivity of the information, and thus increase the set of principals then authorised to access the information.

Combination Algorithm									
ACL	N	N	N	Y	Y	Y	-	-	-
ACL-Capability	N	N	N	-	-	-	Y	Y	Y
Label	N	Y	-	N	Y	-	N	Y	-
Label-Capability	N	-	Y	N	-	Y	N	-	Y
Authorisation Granted/Denied	D	D	D	D	G	G	D	G	G

```
If ((ACL .OR. ACL-Capability) .AND. (Label .OR. Label-Capability))
   Authorisation Granted
Else
   Authorisation Denied
```

Figure 5-9 – Example of Algorithm for Combination of Authorisation Schemes

5.4.5 Management Services

Install ACI

Install ACI establishes an initial set of ACI (for example capabilities for use by initiators, security labels for use by initiator/targets and ACLs for targets) associated with an element. This service may be integral with the creation of an element based upon the security context of the creating activity.

Modify ACI

Modify ACI modifies (for example adds, deletes or changes) the ACI associated with an element

Revoke ACI

Revoke ACI revokes use of ACI relating to an element. This differs from Modify ACI in that any ADI relating to this ACI is revoked.

List ACI

List ACI lists the ACI of a given element

Disable Component

Disable component disables the use of an access control function component. In the case of an AEF component the service inhibits all access through the

AEF.

Re-enable Component
Re-enable component re-enables use of an access control function component.

Within a system there are several sets of such services for each of the distinct security domains comprising the system. As an example there is a set of such services for the platform domain, a set of services for particular applications, for example a RDBMS, and a set of services for support of a common access control service shared by distributed applications, for example OSF DCE.

5.4.6 Operational Services

The following services relate specifically to the exchange and generation of ADI in the formation of associations between a principal and the sponsor of a security domain. These are detailed further in the sections dealing with User identification and Authentication and the Secure Association services.

Acquire Initiator ADI
This service obtains initiator ADI or an access control certificate containing initiator ADI prior to an action.

Acquire Target ADI
This service acquires protected target ADI.

Generate Data ADI
This service binds data ADI to data at the time it becomes available.

Generate Action ADI
This service binds initiator ADI and data ADI to an action necessary for an access control decision to be made.

Revoke ADI
This service revokes the validity of ADI.

Verify Action ADI
This service verifies the validity of action ADI.

Get Contextual Information
This service gets the contextual information required for an access control decision to be made.

Bind Initiator ADI
This service provides for the secure binding of Initiator ADI with a chain of execution within a service domain on an end-system such that the Initiator ADI is securely propagated with the chain of execution.

Retrieve Initiator ADI
This service retrieves the initiator ADI bound to a chain of execution within a security domain on an end-system.

The following service is related to the actual access control mediation process.

Decide Access
This service determines if an access is allowed based on ADI.

5.5 Impact of Access Control Services on Primary Service APIs

5.5.1 Access Control Enforced API / Security Unaware Caller

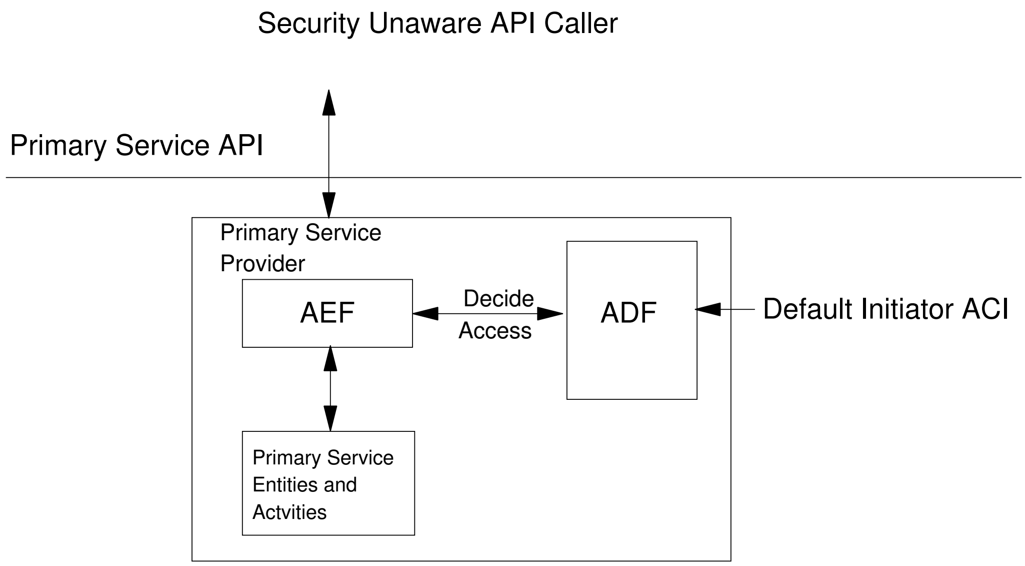


Figure 5-10 – Access Control Enforced API / Security Unaware Caller

In this case the primary service APIs provide no support for the caller of the interface to assume responsibility for access control policy when invoking the API. The access control policy is enforced entirely by the primary service provider. The ADI and contextual information required by the ADF as the basis for the authorisation decision is acquired by the primary service provider (or the ADF). The default Initiator ADI bound to the chain of execution of which invocation of the API forms a part is used as input to the ADF. The primary service provider must implement the AEF but may or may not implement the ADF.

5.5.2 Access Control Enforcing API / Security Selecting Caller

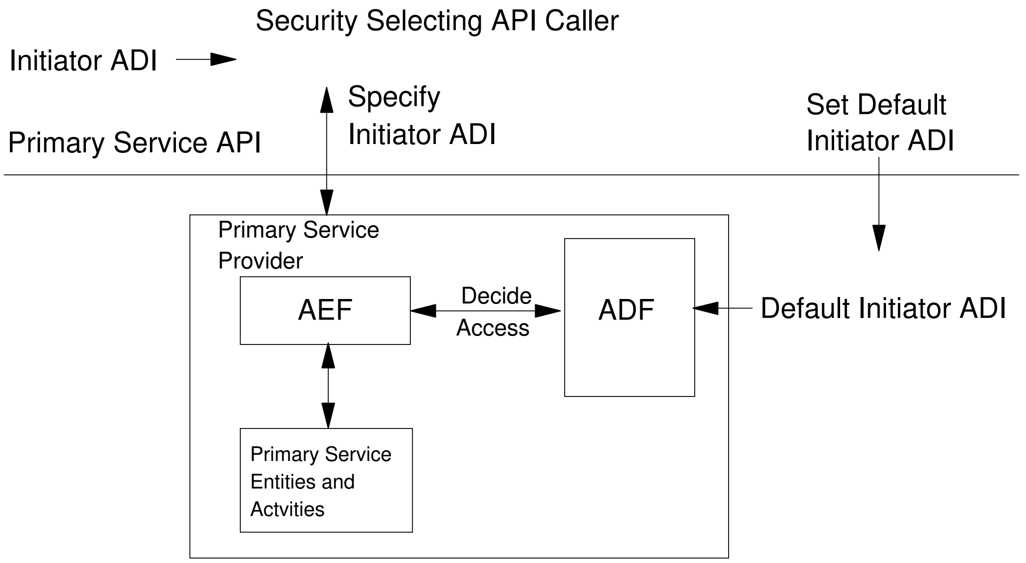


Figure 5-11 – Access Control Enforcing API / Security Selecting Caller

In this case the caller of the API is responsible for selecting the Initiator ADI to be used in the access control decision. This initiator ADI may be passed directly to the primary service provider as parameters to the primary service API or may be provided by means of a separate security service API to bind a particular set of initiator ADI with subsequent primary service invocations as a default. The primary service caller is responsible for managing and ensuring the security of the initiator ADI and its binding to a particular chain of service invocations.

5.5.3 Access Control Non-enforcing API / Access Control Enforcing Caller

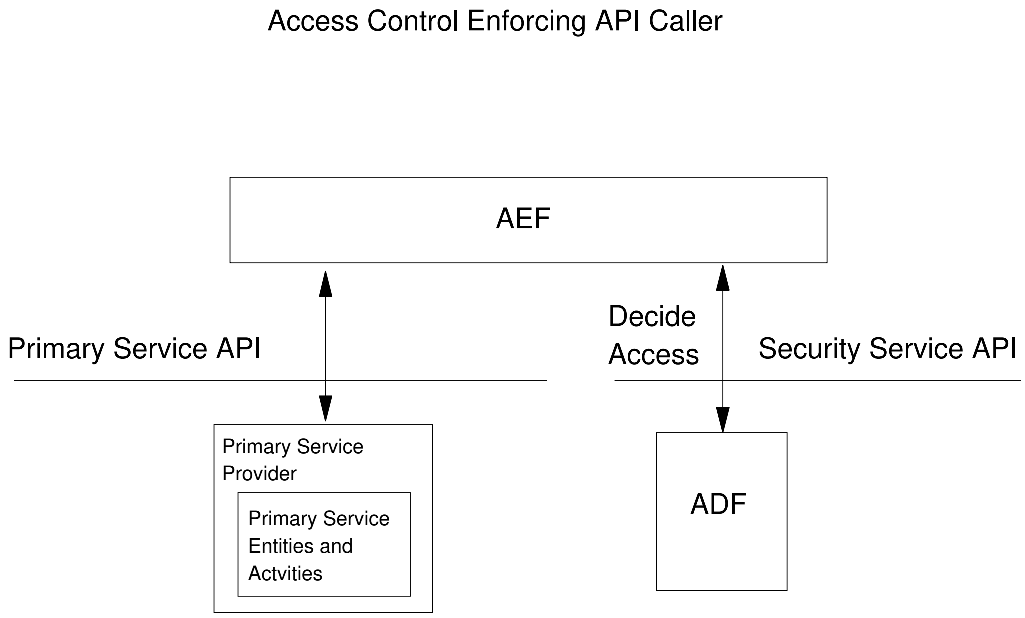


Figure 5-12 – Access Control Non-Enforcing API / Access Control Enforcing Caller

In this case the primary service provider does not enforce access control on the use of its services. The caller of the primary service API is responsible for enforcement of access control policy and therefore contains the AEF. The caller may also contain the ADF or may call a separate service that provides the ADF functionality.

In practice the primary service provider may be required to enforce an access control policy but on different entities from the caller of the primary service. This is to ensure that only callers that are trusted with the responsibility to enforce the appropriate access control policy may invoke the primary service APIs. An example would be a platform service interface that requires the calling process to possess an appropriate capability.

NOTE: This model may also apply if the caller is permitted to assume responsibility for access control policy, in which case the Primary Service Provider permits the caller to override the primary service provider access controls.

5.6 Security Audit Services

5.6.1 Introduction

A security audit service provides for the detection of security relevant actions within an IT system. When such an event occurs, the audit service generates an audit event, which may be recorded, reported, archived, and analysed. Configurable selection criteria determine which events are recorded and reported within a security domain. These criteria are defined by the security policy applicable to the domain.

For their secure working, security audit services place several dependencies on authentication, authorisation and access control measures.

This service description does not encompass a non-repudiation service except to the extent that the recording of security audit information supports such a service.

5.6.2 Security Audit Service Model

Figure xxx illustrates the model of security audit services which comprises the following service components:

Event Detection

The audit event detection service component detects each security relevant activity, and generates an audit event that contains a description of the activity and information from the local security context. Copies of the audit event detection service component are incorporated as needed within primary service providers and security audit detecting callers; see "Impact of Audit Service on Primary Service APIs" below.

Event Discrimination

The audit event discrimination service component receives all audit events and determines their disposition based on configurable criteria and reactions. Depending on which criteria an audit event matches, it may or may not be recorded, and may or may not cause an alarm.

Event Recording

The audit event recording service component receives audit events that are to be recorded. The recording service may add information from the caller's operational security context and the security domain state to the audit event. The event is added to the appropriate audit trail as specified by a configurable destination.

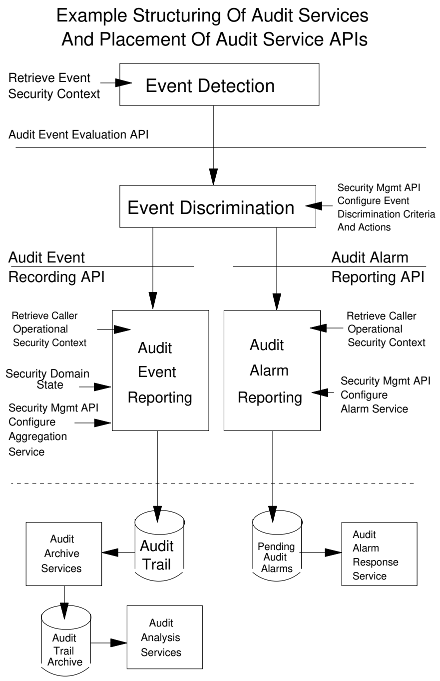


Figure 5-13 – Basic Security Audit Service Model

Alarm Reporting

The audit alarm reporting service component receives audit events that are to cause an alarm. An event that is configured as an alarm event must either immediately initiate an alarm or be added to a collection of such events to subsequently trigger an alarm when a configured threshold of events has occured. The alarm reporting service may add information from the caller's operational security context as it generates an alarm from the audit event. When triggered, an alarm is sent to a configurable destination.

The triggering of an alarm may also initiate an action to protect the system from further threat, for example by inhibiting the activity causing the generation of the audit events that are triggering the alarm. An example may be the invocation of an Invalidate AI service or Invalidate ACI service, or both to prevent further use of a principal identity or I/O device in the case of failed or duplicated authentication operations.

Alarm Examination

Audit alarms that have been generated have to be examined and responded to by a representative of the security authority of the security domain, usually the security auditor.

Audit Archiving

The audit archiving service component allows audit trails to be saved for later analysis.

Audit Analysis

The audit analysis service component retrieves and merges archived audit trails and produces reports.

5.6.3 Distributed Audit Services

Within this Security Framework an IT system comprises many security domains. An audit service is generally supported as one or more specialist security domains providing all or parts of the total audit service to all, or a subset of, the other security domains.

The specific activities considered security relevant within each security domain and therefore requiring auditing vary according to the function of the security domain. Additionally if activities span security domains under the ultimate control of different security authorities, then some mapping of security attribute syntax and semantics may be required.

In order to support the centralised archiving and analysis of audit trails in a distributed heterogeneous environment it is necessary to establish a common audit trail interchange format, including a common interpretation of audit event types.

A common interpretation of audit event types is best achieved by considering security relevant actions at the abstract level. Examples of such abstract actions may comprise:

- association creation and termination
- security relevant entity creation and deletion
- security management actions (for example install, deinstall, disable, re-enable).

An audit trail export function is then required to convert any local representation of an audit trail to such a common interchange and analysis format.

NOTE: A centralised audit analysis service may provide an analysis at a suitably abstract level. If security exceptions are identified, a lower level of audit analysis may be undertaken. Eventually it may be necessary to revert to source machine format for a detailed analysis.

An interchange format may comprise a common interchange header and detail record together with an opaque source format record.

5.6.4 Management Services

• Configure Audit Event Discrimination

— install audit event discrimination criteria and reactions

Installs the criteria used to control which audit events are to be generated and the consequence action. For example, criteria may include:

— type of entity performing the action

— sensitivity of the action

— attributes of the information being processed

— context of the action, for example location, and time of day (allowing for higher level of recording out of normal working hours).

Examples of consequent action include no reaction or generate alarm or record event or record event and generate alarm.

— modify audit event discrimination criteria and reactions

Modifies the criteria and actions used in event discrimination.

— deinstall audit event discrimination criteria and actions

Removes selected criteria and actions so they no longer affect event discrimination.

• Configure Audit Alarm Reporting

— install audit alarm

Installs instructions that determine what happens when alarms are received. This requires the specification of such information as threshold values, destinations of alarm message, contents of alarm message, actions to be taken in addition to sending alarm (eg., disable user account and terminal).

— modify audit alarm

Modifies what happens when alarms are received.

— deinstall audit alarm

Removes instructions for selected alarms.

— disable/re-enable audit alarm

Temporarily turns off or on alarm reporting for a selected alarm.

• Configure Audit Event Recording

— install audit trail

Initializes an audit trail and designates it as the destination for recorded audit events.

— deinstall audit trail

Stops sending events to the audit trail.

— disable/re-enable audit trail

Temporarily stops or restarts adding events to the selected audit trail.

• Audit Archiving

— archive audit trail

Saves an audit trail for long term storage. The audit trails need to be securely archived in order to allow for subsequent analysis, possibly after a long elapsed time.

— export audit trail

Converts an audit trail to standard interchange format and copies it to a given destination.

• Audit analysis

— import audit trail archive

Retrieves for analysis a copy of an audit trail that was previously archived.

— merge audit trail

Allows the merging of several audit trails and the tracing of chains of events.

— produce reports

Provides measures for analysing the audit trails produced and generating useful reports on security relevant usage and activity. Procedures, mechanisms and tools are required to ensure that records of security-sensitive events are regularly analysed. Similarly, procedures and mechanisms are required to ensure that business-sensitive events are subsequently available to support the resolution of legal and contractual disputes.

5.6.5 Operational Services

• Audit Event Detection

The detection of actions must occur within the software that provides the services that constitute the security relevant activity. That is, action detection is part of the primary service provision and must be included within the specification of that service provision. This means that an API specification for the primary service shall include any requirement for the generation of an audit event.

The audit events generated must include an assertion of the identity of the presenter of the event.

• Audit Event Discrimination

Audit event detection has to be included as a fundamental part of a primary service provision. However, the security policy may permit the security authority, as represented by the Security Auditor, to configure the actions to be taken on the detection of an audit event to be based on local or current context. A received event therefore has to be screened against the actions configured for the event.

Audit event discrimination must verify that the events it receives are from valid sources. This will prevent attacks that attempt denial of service by overloading the security audit services.

• Audit Event Recording

An event that requires recording has to be appended to an audit trail in a manner such that it cannot be subsequently modified or deleted. For each event the identity of the initiating entity or performing entity or both, together with other pertinent descriptors of the event, for example date and time, are recorded.

Other information that must be recorded includes the operational security context of the event together with such security domain management information that permits the record to be interpreted (for example, mappings of internal security attribute values to text strings). This may include the recording of cryptographic keys used for enciphering or sealing data. For efficiency purposes the security domain management information may be recorded periodically (for example, once per day) and the security management state applicable to any individual event regenerated by tracing all audited changes to that information since the last record of the state.

• Audit Alarm Reporting

An event that is configured to generate an alarm must either immediately initiate an alarm or be added to a collection of such events to subsequently trigger an alarm when a configured threshold of such events has occurred. The speed with which the alarm is raised is determined by the severity of the event. Ideally, the reporting of the alarm should give as much time as possible to those responsible for limiting the potential damage. The alarms are reported over one or more robust communication services to designated user interfaces.

- Audit Alarm Response

Audit alarms that have been generated have to be examined and responded to by a representative of the security authority of the security domain, usually the Security Auditor.

- suspend primary service auditing

(See xxx below.)

5.6.6 Impact of Audit Service on Primary Service APIs

Security Audit Detecting API / Security Unaware Caller

If a primary service provider supports services that may be interpreted as security auditable actions by the security policy then the primary service provider is required to incorporate the audit action detection capability and to invoke the other audit services if an auditable action is detected. The specification of the primary service APIs should include the requirement to audit the API. The caller of the security action may be unaware of any auditing requirements. The primary service implementation, or the audit services it invokes, must derive all necessary security context information for inclusion in the audit record.

Security Audit Detecting Caller

If the caller of a primary service API also provides services that are interpreted as auditable actions then it is also required to incorporate audit event generation functionality and to invoke the other audit services. In this case the security policy may permit the caller of the primary service API to modify the event discrimination configuration for the primary service to suppress the generation of any audit records or alarms from its use of the primary service.

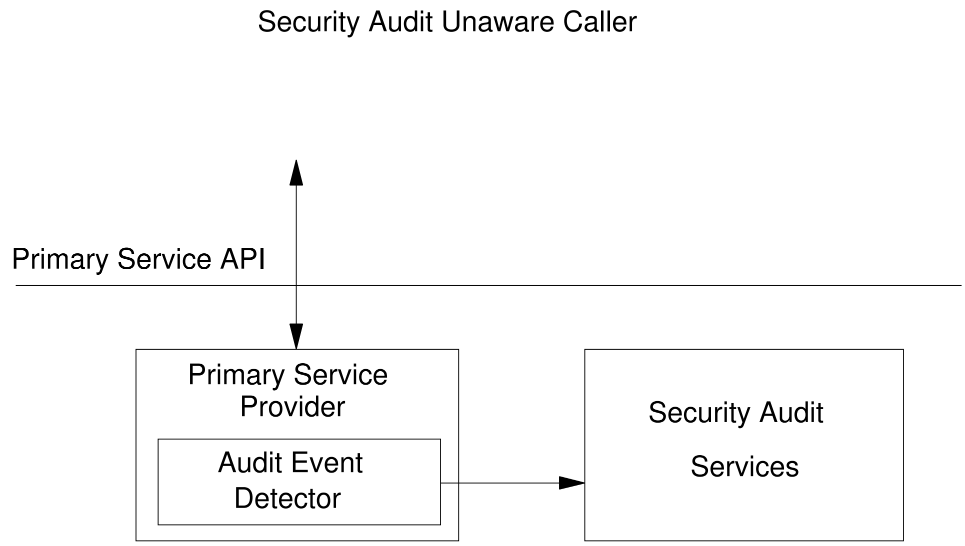
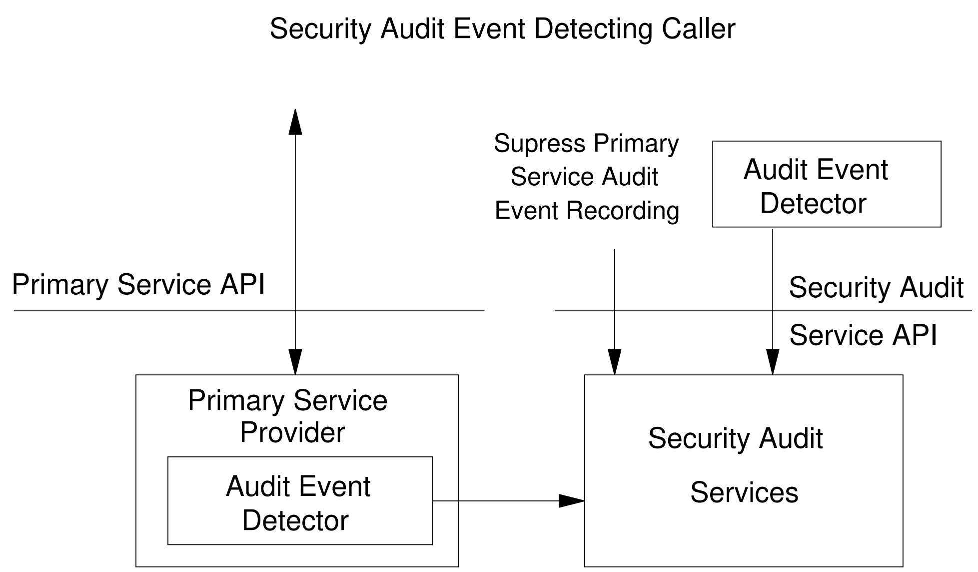


Figure 5-14 – Security Audit Detecting API / Security Unaware Caller



1
2

3 **Figure 5-15 – Security Audit Detecting Caller**