Guide

Architecture for Public-Key Infrastructure (APKI)

Draft 1

The Open Group

Guide

Architecture for Public-Key Infrastructure (APKI) Draft 1

Document Number:

Published in the U.K. by The Open Group, May 1997.

Any comments relating to the material contained in this document may be submitted to:

The Open Group
Apex Plaza
Forbury Road
Reading
Berkshire, RG1 1AX
United Kingdom

or by Electronic Mail to:

OGSpecs@opengroup.org

Contents

List of Figures

Preface

The Open Group

The Open Group is an international open systems organization that is leading the way in creating the infrastructure needed for the development of network-centric computing and the information superhighway. Formed in 1996 by the merger of the X/Open Company and the Open Software Foundation, The Open Group is supported by most of the world's largest user organizations, information systems vendors and software suppliers. By combining the strengths of open systems specifications and a proven branding scheme with collaborative technology development and advanced research, The Open Group is well positioned to assist user organizations, vendors and suppliers in the development and implementation of products supporting the adoption and proliferation of open systems.

With more than 300 member companies, The Open Group helps the IT industry to advance technologically while managing the change caused by innovation. It does this by:

- consolidating, prioritizing and communicating customer requirements to vendors
- conducting research and development with industry, academia and government agencies to deliver innovation and economy through projects associated with its Research Institute
- managing cost-effective development efforts that accelerate consistent multi-vendor deployment of technology in response to customer requirements
- adopting, integrating and publishing industry standard specifications that provide an essential set of blueprints for building open information systems and integrating new technology as it becomes available
- licensing and promoting the X/Open brand that designates vendor products which conform to X/Open Product Standards
- promoting the benefits of open systems to customers, vendors and the public.

The Open Group operates in all phases of the open systems technology lifecycle including innovation, market adoption, product development and proliferation. Presently, it focuses on seven strategic areas: open systems application platform development, architecture, distributed systems management, interoperability, distributed computing environment, security, and the information superhighway. The Open Group is also responsible for the management of the UNIX trade mark on behalf of the industry.

The X/Open Process

This description is used to cover the whole Process developed and evolved by X/Open. It includes the identification of requirements for open systems, development of CAE and Preliminary Specifications through an industry consensus review and adoption procedure (in parallel with formal standards work), and the development of tests and conformance criteria.

This leads to the preparation of a Product Standard which is the name used for the documentation that records the conformance requirements (and other information) to which a vendor may register a product. There are currently two forms of Product Standard, namely the Profile Definition and the Component Definition, although these will eventually be merged into one.

The X/Open brand logo is used by vendors to demonstrate that their products conform to the relevant Product Standard. By use of the X/Open brand they guarantee, through the X/Open Trade Mark Licence Agreement (TMLA), to maintain their products in conformance with the Product Standard so that the product works, will continue to work, and that any problems will be fixed by the vendor.

Open Group Publications

The Open Group publishes a wide range of technical literature, the main part of which is focused on specification development and product documentation, but which also includes Guides, Snapshots, Technical Studies, Branding and Testing documentation, industry surveys and business titles.

There are several types of specification:

- *CAE Specifications*

  CAE (Common Applications Environment) Specifications are the stable specifications that form the basis for our product standards, which are used to develop X/Open branded systems. These specifications are intended to be used widely within the industry for product development and procurement purposes.

  Anyone developing products that implement a CAE Specification can enjoy the benefits of a single, widely supported industry standard. In addition, they can demonstrate product compliance through the X/Open brand. CAE Specifications are published as soon as they are developed, so enabling vendors to proceed with development of conformant products without delay.

- *Preliminary Specifications*

  Preliminary Specifications usually address an emerging area of technology and consequently are not yet supported by multiple sources of stable conformant implementations. They are published for the purpose of validation through implementation of products. A Preliminary Specification is not a draft specification; rather, it is as stable as can be achieved, through applying The Open Group's rigorous development and review procedures.

  Preliminary Specifications are analogous to the *trial-use* standards issued by formal standards organizations, and developers are encouraged to develop products on the basis of them. However, experience through implementation work may result in significant (possibly upwardly incompatible) changes before its progression to becoming a CAE Specification. While the intent is to progress Preliminary Specifications to corresponding CAE Specifications, the ability to do so depends on consensus among Open Group members.

- *Consortium and Technology Specifications*

  The Open Group publishes specifications on behalf of industry consortia. For example, it publishes the NMF SPIRIT procurement specifications on behalf of the Network Management Forum. It also publishes Technology Specifications relating to OSF/1, DCE, OSF/Motif and CDE.

  Technology Specifications (formerly AES Specifications) are often candidates for consensus review, and may be adopted as CAE Specifications, in which case the relevant Technology Specification is superseded by a CAE Specification.

In addition, The Open Group publishes:

- *Product Documentation*

  This includes product documentation — programmer's guides, user manuals, and so on — relating to the Pre-structured Technology Projects (PSTs), such as DCE and CDE. It also includes the Single UNIX Documentation, designed for use as common product documentation for the whole industry.

- *Guides*

  These provide information that is useful in the evaluation, procurement, development or management of open systems, particularly those that relate to the CAE Specifications. The Open Group Guides are advisory, not normative, and should not be referenced for purposes of specifying or claiming conformance to a Product Standard.

- *Technical Studies*

  Technical Studies present results of analyses performed on subjects of interest in areas relevant to The Open Group's Technical Program. They are intended to communicate the findings to the outside world so as to stimulate discussion and activity in other bodies and the industry in general.

- *Snapshots*

  These provide a mechanism to disseminate information on its current direction and thinking, in advance of possible development of a Specification, Guide or Technical Study. The intention is to stimulate industry debate and prototyping, and solicit feedback. A Snapshot represents the interim results of a technical activity.

Versions and Issues of Specifications

As with all *live* documents, CAE Specifications require revision to align with new developments and associated international standards. To distinguish between revised specifications which are fully backwards compatible and those which are not:

- A new *Version* indicates there is no change to the definitive information contained in the previous publication of that title, but additions/extensions are included. As such, it *replaces* the previous publication.
- A new *Issue* indicates there is substantive change to the definitive information contained in the previous publication of that title, and there may also be additions/extensions. As such, both previous and new documents are maintained as current publications.

Corrigenda

Readers should note that Corrigenda may apply to any publication. Corrigenda information is published on the World-Wide Web at **http://www.opengroup.org/public/pubs**.

Ordering Information

Full catalogue and ordering information on all Open Group publications is available on the World-Wide Web at **http://www.opengroup.org/public/pubs**.

This Document

This document is a Guide (see above).

- Chapter 1 describes the requirements on a Public-Key Infrastructure.
- Chapter 2 presents the high-level structure of the PKI Architecture by grouping the architecture's components into broad functional categories.
- Chapter 3 on page 111
  - enumerates the components in each of the Architecture's functional categories
  - describes the functionality of each component and lists existing specifications which could serve as candidate standards for each component's interfaces and protocols (To be considered a "candidate" for purposes of the public-key infrastructure architecture, an interface or protocol must:
    1. be described by a publicly-available specification, and
    2. support a significant fraction of the functionality of the PKI component for which it is proposed as a candidate.

    It is assumed that the candidate interface and protocol specifications identified in this document will serve as base documents for open standardization processes, which will produce finalized PKI component interface and protocol specifications.)
  - identifies where negotiation facilities are required to deal with the probable existence of a multiplicity of security mechanisms
  - enumerates important public-key-related protocols and discusses the need for environment-specific profiles
- Chapter 4 discusses the use of hardware security devices in the architecture.
- A glossary and index are provided.

The Open Group PKI TG continues to refine and extend these requirements; comments should be sent by electronic mail to pki-tg@opengroup.org.

Trade Marks

OSF™ is a trademark of The Open Software Foundation, Inc.

X/Open® is a registered trademark, and the ‘‘X’’ device is a trademark, of X/Open Company Limited.

Acknowledgements

The OpenGroup gratefully acknowledges the work of the OpenGroup Security Program Group in the development of this specification and the following individuals:

Anne Anderson (HP)	Charles Blauner (JP Morgan)
Belinda Fairthorne (Fujitsu-ICL)	Warwick Ford
Robert Jueneman (Novell)	Ellen McDermott (Open Market)
Howard Melman (OSF)	Denis Pinkas (Groupe Bull)
Walt Tuvell (OSF)	John Wray (Digital Equipment Corporation)

Additionally, the following organisations contributed to the specification of the requirements.

Amdahl	Barclays Bank
BCTEL	Bellcore
Boeing	Bull
Citicorp	Digital Equipment Corporation
Dynasoft	Electronic Data Systems
Fujitsu-ICL	Guide International
Harris Corporation	HP
IBM	Information & Support Group
Jet Propulsion Laboratory	J P Morgan
Lockheed Martin	Mitre
NCR	NIST
Nortel	Pacific Gas & Electric
SCO	Shell International
Siemens Nixdorf	SUN
Sweden Post	Telecom Finland Ltd.
The Open Group	UK Ministry of Defense
US DISA	US NSA
Veritas	

Referenced Documents

The following documents are referenced in this specification:

ISO / IEC 7498-2

ISO / IEC 7498-2: 1989, Information Processing Systems — Open Systems Interconnection — Basic Reference Model — Part 2: Security Architecture.

X.509

ISO / IEC 9594-8: 1990, Information Technology — Open Systems Interconnection — The Directory — Part 8: Authentication Framework, together with:

Technical Corrigendum 1: 1991 to ISO / IEC 9594-8: 1990.

ECMA TR / 46

Security in Open Systems, A Security Framework, July 1988, European Computer Manufacturers Association.

draft-ietf-pkix-ipki-part1-04.txt

This document describes profiles for use of X.509 certificates and certificate revocation lists (CRLs) and their respective extension fields in the Internet environment

draft-ietf-pkix-ipki3cmp-01.txt

This document describes protocols for certificate management in the Internet environment

Internet RFC 1508

This document describes the GSS-API interface, which provides integrity and privacy services for session- oriented messages

draft-ietf-wts-gssapi-00.txt.

This document describes how to use GSS-API to protect Web transactions (HTTP protocol exchanges, in particular)

draft-ietf-cat-idup-gss-07.txt

This document describes the IDUP-GSS-API interface, which provides integrity and privacy services for store-and- forward messages, and non-repudiation services.

IETF RFC 2025

This document describes how to use the SPKM protocol under a GSS-API interface

draft-ietf-cat-sesamemech-02.txt

This document describes the use of the SESAME protocols under a GSS-API interface.

draft-ietf-cat-snego-04.txt

This document describes a proposed mechanism negotiation preamble protocol for use by protocol partners wishing to use GSS-API to establish a secure association.

draft-ietf-pkix-ipki2opp-00.txt

This document describes protocols for retrieving certificates and CRLs in an Internet environment.

draft-ietf-pkix-ipki-part4-00.txt

This document describes a standard certification policy and certification practices for the Internet environment.

The SSL Protocol v3

Describes version 3 of the SSL protocol; available from Netscape Web site

The following X/Open documents are referenced in this specification:

Base GSS-API

CAE Specification, December 1995, Generic Security Service API (GSS-API) Base (ISBN: 1-85912-131-4, C441).

XDSF

Guide, December 1994, Distributed Security Framework (ISBN: 1-85912-071-7, G410).

Chapter 1

Requirements on a Public Key Infrastructure

1.1 Baseline Requirements for a Global PKI

An interoperable global PKI is required to provide privacy and digital signature services in support of international commerce, balancing the legitimate needs of commerce, governments and privacy of citizens. The global PKI must support multiple governance policy models within a single global PKI framework, and must enable the enforcement of all existing governance policy mandates.

1.1.1 Required Services

- Establishment of domains of trust and governance
- Confidentiality (sealing)
- Integrity and authentication (signing)
- Non-repudiation
- End-to-end monitoring, reporting and auditing of PKI services

1.1.2 Required Functionality and Characteristics

Key life-cycle management

The actual life cycle of a key depends on whether it is used for confidentiality or signature purposes. Key life-cycle facilities to be supported are:

1. Key recovery facilities

   The PKI shall specify key recovery functionality for use in environments which require such functionality. This document takes no position on key recovery policy issues. Implementations of the PKI may omit key recovery functionality, or may disable its use, in environments in which it is not required. PKI implementations which provide key recovery functionality should do so using the interfaces and/or protocols specified herein. Key recovery facilities shall provide the following functionality:

   - Use of key recovery facilities implies acceptance of a mandatory policy for the protection and recovery of keys. The policy defines how the keys are to be protected and under what conditions and to whom a key will be made available. The mandatory aspect of policy arises as the operations of a key recovery facility may be regulated by legislation or procedures required under commercial contracts for liability management.
   - It must be possible to insure that only key recovery enabled systems shall be usable within a PKI implementation, where this is required.

- A key recovery facility shall be unconditionally trusted and be liable to uphold the stated policy with redress for loss arising from failures to uphold policy through contractual liability and penalties.
- A key recovery center shall be able to verify the legitimacy of a key submitted to it for storage.
- A user of a key recovery repository shall be able to verify that it is an authorized repository.
- The PKI shall provide for coordination between the management of public and private keys in PKI and in data recovery centers.

  Note: Public and private key parts do not have the same life cycle and key parts may be archived.
- The PKI shall support aging, revocation, and repudiation of keys.
- The PKI shall support discretionary key fragmentation between key recovery facilities.

2. Key generation facility

   The method of key generation shall be discretionary, subject to commercial decision and business requirement. Selection of key quality, uniqueness, secrecy and recoverability of keys must be left to the discretion of the organization generating the keys (and any governance authorities to which it is subject).

3. Key Distribution, Revocation, Suspension, Repudiation and Archive

   The PKI must support the following functionality:

   - Facilities for the distribution of keys to appropriate storage devices and directories.
   - Ability of a certification authority to revoke certificates for individual keys under the terms of the applicable policy.
   - Ability of a certification authority to suspend and reactivate certificates for individual keys under the terms of the applicable policy.
   - Ability of a certification authority to force delivery of revocation, suspension, and reactivation notices.
   - Facilities to enable a user to repudiate his public key under the terms of the applicable policy.
   - Facilities to enable a user to suspend and reactivate his public key under the terms of the applicable policy.
   - Facilities to enable the user and subscriber to retrieve revocation, suspension, and reactivation notices.
   - Facilities to enable the user and subscriber to determine the status (e.g., revoked or suspended) of a specific certificate.
   - Facilities to enable the archive and subsequent retrieval of certificates in support of the retrieval and verification of long term information in accordance with governance policy.
   - Warranted retrieval

     The PKI must support implementations which enable the following warranted retrieval scenarios:

- Law enforcement retrieval (subject to policy conditions)
- Corporate agency retrieval (subject to policy and authorizations)
- Individual retrieval (subject to policy and authorizations)

The following functionality is required in support of warranted retrieval:

- An electronic vehicle for the delivery of a notarized electronic warrant, to support the automation of key retrieval under due process (this must be able to take advantage of existing legal agreements)
- A permanent, non-repudiable and independently verifiable record of key retrieval operations must be maintained.

Note: Warranted retrieval policy includes policy regarding disclosure or non-disclosure of key retrieval to owner of the retrieved key.

Distributed Certificate Management Structure

The PKI must provide distributed Certificate Management functionality, driven by the requirements of the transaction or business domain. The following Certificate Management functions must be provided by the PKI:

1. Policing and policy enforcement (governance model), including the following:
   - Policy creation and maintenance. The policies include those covering key generation, key recovery, key distribution, revocation, suspension, repudiation, archive and warranted retrieval.
   - Ability to register a key and the binding between the key and a name.
   - Ability to query which keys are bound to a name
   - Policies (for services built on PKI access control) must not be required to be based on individual identity.
   - Certification of the binding between a public key and a directory name shall be mandatory
   - Certification of the binding between additional attributes and a directory name shall be discretionary
   - Auditing and support for the monitoring of policy compliance is required
2. Concurrent support of multiple policies
3. exchange of certificates.
4. Support for continuance of service in the event of transfer of certificate services from one certification authority to another.
5. Certificate authority policy mapping services to establish cross certification between CAs.
6. Support for arbitration to determine acceptability of certificates in the event of multiple conflicting certification paths.
7. Support for separation of the certification authority and repository functions in accordance with the governance policy. changes to certificate repositories must be transactional (e.g., two-phase commits).

Security of the PKI

The PKI itself must be secure. In particular, the PKI must:

1. Protect the confidentiality, integrity and availability of the PKI services, for example key generation, key distribution, and key storage.
2. Provide strong non-repudiation services for actions of certificate services.
3. Prevent PKI services themselves from repudiating their own actions.
4. Prevent users and subscribers from repudiating their own actions.

Time service

A universal, networked time service must be available for time stamping.

Interoperability

PKI elements provided by different vendors must interoperate. In support of interoperability, PKI elements must:

1. support international standards for certificates and associated data
2. support international standards for certificate services
3. support internationalization of all certificates and associated data
4. support internationalization of all certificate services

1.1.3 Known Issues

For interoperability there is a dependency upon the definition of standard application program interfaces to and protocols between the component services of the Public Key Infrastructure.

Work is required to define and agree profiles of option fields in certificates.

1.1.4 Recommendations

Adopt X.509 version 3 as a basis for certificates in the development of the PKI.

Adopt and adapt existing standards and protocols wherever possible, invent new standards or protocols only as a last resort.

1.2 The Importance of Architecture

The APKI working group feels that a robust, flexible, standard, open Public-Key Infrastructure Architecture is critical to the success of secure systems based on Public-Key technology. This section explains why.

1.2.1 What is Architecture?

The architecture of a software system is the set of interfaces through which its functions are accessed, and the set of protocols through which it communicates with other systems.

The remainder of this section discusses the importance of standardizing the interfaces and protocols which comprise the Public-Key Infrastructure software Architecture.

1.2.2 Interfaces

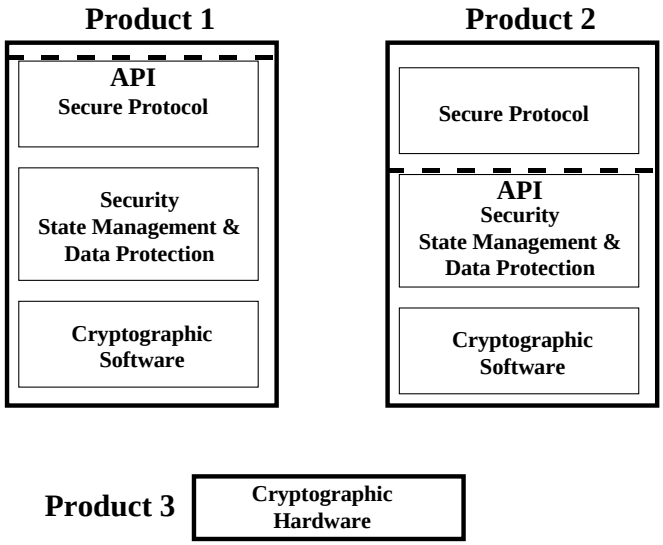


Figure 1-1 Example Security Products

Figure 1-1 illustrates a system on which three security products have been installed.

In the figure:

- Product 1 includes a protocol and all the security functionality needed to protect data flowing over that protocol. Only the secure protocol's interface is exposed; the underlying security functionality is not available to other applications.
- Product 2 also includes a protocol and its requisite security functionality, but it exposes the data protection functionality through a public interface so that other applications can use it. It does not permit direct access to cryptographic functionality.
- Product 3 is a hardware cryptographic adapter; it comes with a software driver permitting access by applications to its cryptographic functionality.

This configuration has several bad characteristics:

- Because neither product 1 nor product 2 accesses cryptographic functionality through a standard interface, neither can use the cryptographic adapter. Furthermore, because both product 1 and product 2 embed cryptographic functionality without exposing an interface through which it can be accessed, neither can use the other's cryptographic software. The end result is that three different cryptographic subsystems (two software and one hardware) must be installed on the system, even if all three products use the same cryptographic algorithms!
- Because product 1 and product 2 embed cryptographic functionality rather than accessing a separate cryptographic subsystem through a published interface, they will not be deployable (without code changes) in countries whose regulatory environment restricts or forbids use of the cryptographic functions they embed.

This example illustrates some of the benefits of standard interfaces; these include:

- Replaceability of services (e.g. cryptography) without change to exploiting applications
- Elimination of duplicate service implementations in configurations in which multiple applications require the same kind of service
- Reduced programmer training costs (programmers need learn only one standard interface for a service rather than learning the proprietary interfaces of multiple products providing the same service)
- Reduced application porting complexity (code exploiting services through standard interfaces need not be changed, or requires only minimal changes, when porting from one platform supporting the standard interface to another such platform)

1.2.3 Protocols

Figure 1-2 illustrates two certificate- management products.

In the figure:

- Product 1 communicates key requests to the Certification Authority (CA) via electronic mail, and receives keys and certificates from the CA via email.
- Product 2 communicates key requests to the CA using a proprietary protocol and retrieves keys from a directory service using the LDAP protocol.

A configuration including both products would have several bad characteristics:

- Neither product's CA could accept key requests from the other product's clients.
- Applications using product 1 clients and wishing to advertise their certificates in the directory service would require installation of a separate directory- access product.
- Applications using product 1 clients and wishing to retrieve partners' certificates from the directory service would require installation of a separate directory-access product.

This example illustrates the benefit of standard protocols:

- Applications supporting standard protocols can interoperate, even if produced by different providers.

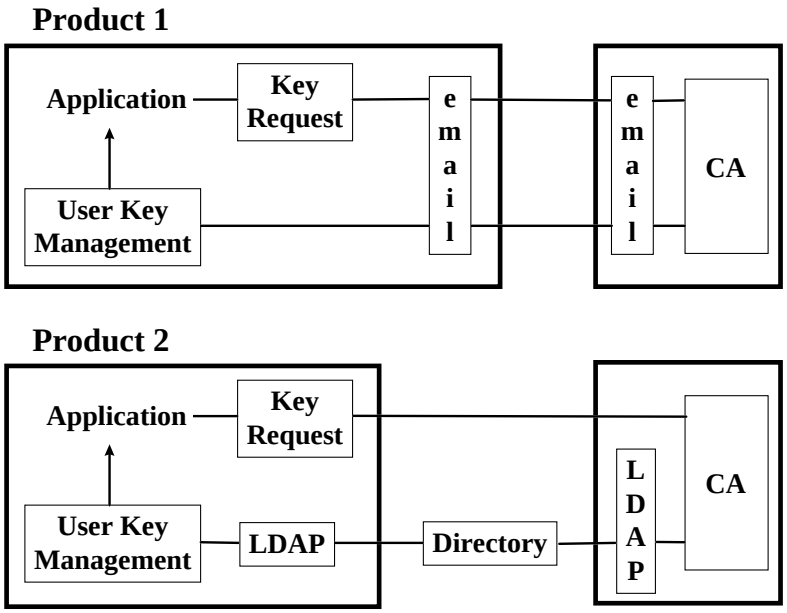


Figure 1-2 Protocols in Certificate Management

1.2.4 Profiles

Many of the services in the Public-Key Infrastructure Architecture can be implemented using a variety of different mechanisms and protocols (e.g. data privacy protection can be implemented using a variety of different cryptographic algorithms). This variety of mechanisms and protocols has arisen in part because different environments impose different security requirements.

Multiplicity of mechanisms means that different providers' implementations of the PKI Architecture will not necessarily interoperate - even though they support the standard interfaces and a selection of the standard protocols.

A profile defines the set of mechanisms and protocols which should be used in a particular environment. The mechanisms and protocols comprising a profile are usually chosen on the basis of their strength against the attacks which are common in the environment supported by the profile. Profiling has the following advantages:

- Systems conforming to an environment's profile will interoperate.
- Systems conforming to an environment's profile will be well-protected against that environment's risks.
- Profiling helps to assure that mechanisms in use work together appropriately and securely.

1.2.5 Negotiation

Some profiles will allow multiple mechanisms and protocols in order to support different qualities of protection, or to accommodate a fragmented security product market. In these environments, it is desirable to provide a negotiation meta-protocol which allows communicating partners to determine:

- which mechanisms and protocols they both (or all) share
- which mechanism and protocol, among the shared set, best supports the desired quality of protection.

Note: It is important to note that negotiation does not always require an on-line dialog between the negotiating entities.

Chapter 2

Overview of the PKI Architecture

The PKI architecture components are grouped into the following broad functional categories:

- System Security Enabling Services provide the functionality which allows a user's or other principal's identity to be established and associated with his actions in the system.
- Crypto Primitives and Services provide the cryptographic functions on which public-key security is based (including secret-key primitives such as DES).
- Long-term Key Services permit users and other principals to manage their own long-term keys and certificates and to retrieve and check the validity of other principals' certificates
- Protocol Security Services provide security functionality (data origin authentication, data integrity protection, data privacy protection, nonrepudiation) suitable for use by implementors of security-aware applications such as secure protocols.
- Secure Protocols provide secure inter-application communications for security-unaware and "mildly" security-aware applications.
- Security Policy Services provide the policy-related information which must be carried in secure protocols to enable access control, and provide access-control checking facilities to security-aware applications which must enforce policy.
- Supporting Services provide functionality which is required for secure operation, but is not directly involved in security policy enforcement.

Figure 2-1 illustrates the PKI architecture.

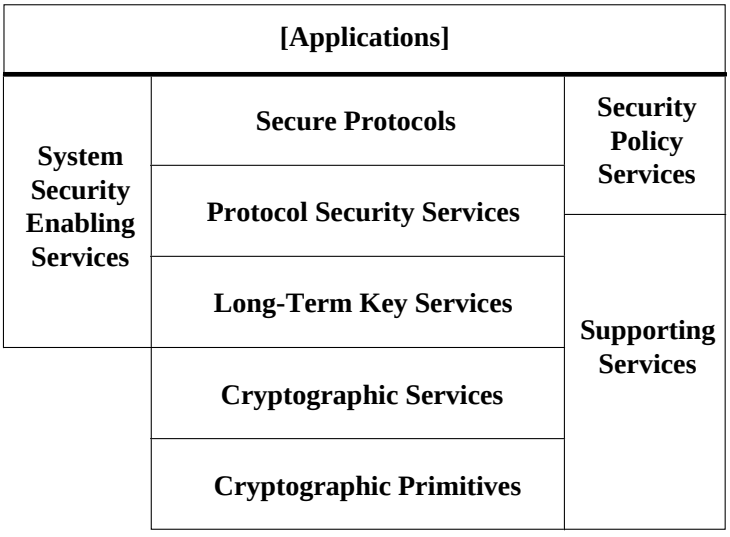


Figure 2-1 PKI Architecture Overview

Chapter 3 describes each of these categories in more detail (listing the components in each category), and identifies interfaces and protocols which may be candidate bases for standardization of each component.

Note: While the architecture described in this document could be implemented on insecure operating system platforms, implementors of the architecture must insure that keys, security context data, and policy data are appropriately protected in such environments.

Chapter 3

Public-Key Infrastructure Components

Figure 2-1 outlined the functional categories comprising the PKI Architecture and showed their relationship in the diagram repeated as Figure 3-1.

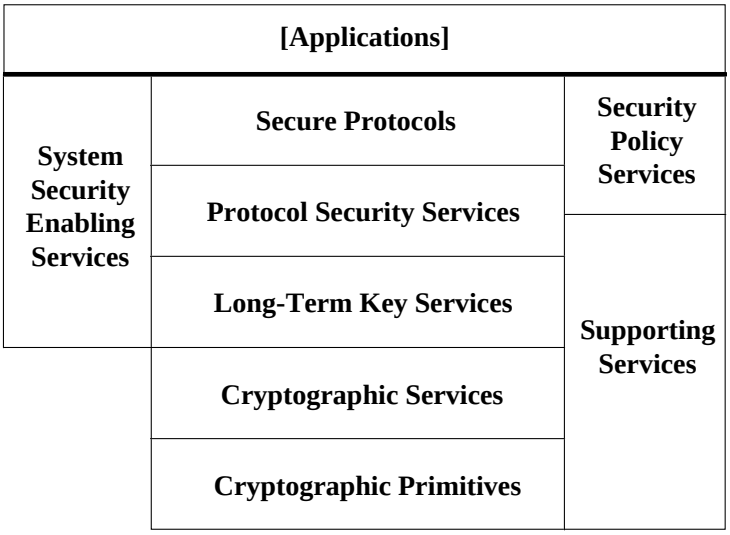


Figure 3-1 PKI Architecture

Each of this section's subsections describes one of the Architecture's categories in detail, enumerating its components and describing component functions, interfaces, and protocols.

3.1 Crypto Primitive Components

Figure 3-2 illustrates the Crypto Primitive Components:

Note: The architecture's cryptographic primitives may be provided by hardware (e.g. smartcards or cryptographic modules) or by software.

3.1.1 Function

These components provide access to low-level cryptographic primitives such as key generation, hash function application to a data buffer, encryption of a data buffer using secret-key or public-key algorithms, decryption of a data buffer using secret-key or public- key algorithms, etc....

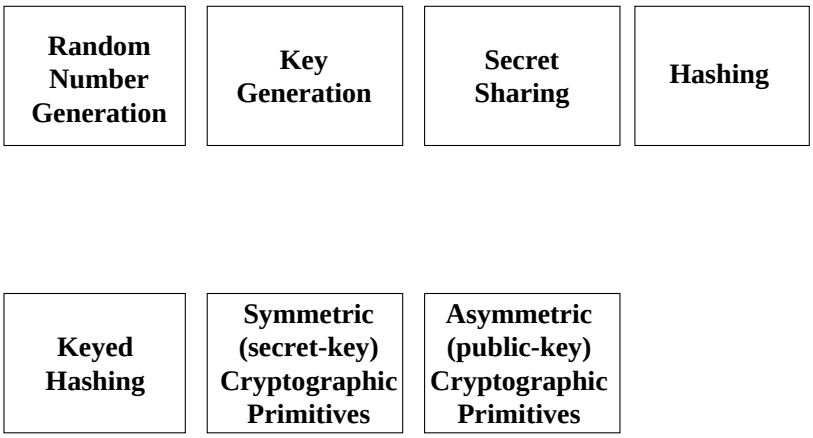


Figure 3-2 Cryptographic Primitive Components

3.1.2 Protocols

Cryptographic primitives are typically called locally; it is not anticipated that any cryptographic primitive protocols will be defined.

3.1.3 Interfaces

Candidate interfaces for access to cryptographic primitives include:

- The RSA BSafe library interface
- RSA PKCS-11
- The X/Open GCS-API
- The Microsoft CryptoAPI 1.0

Other interfaces which may support some or all of the cryptographic primitive function include

- Fortezza
- IBM CCA

Standardization of these interfaces would be of interest to developers of cryptographic service modules and to providers of cryptographic primitive modules. Standardization of an interface for access to cryptographic primitives would facilitate "pluggable" implementations of cryptographic services. The consensus of the APKI working group, however, is that cryptographic functionality will ordinarily be used through the cryptographic service interfaces rather than through the cryptographic primitive interfaces. Therefore, standardization of cryptographic primitive interfaces is not viewed as essential.

3.1.4 Profiles

Most cryptographic modules provide support for multiple primitives. Many primitives are subject to legal restrictions on deployment (including both intellectual property encumbrances and national and international regulatory constraints on export, import, and deployment).

Cryptographic primitive profiles will have to be developed for PKI environments of interest (including, for example, the Internet, OMG CORBA, OSF DCE, Financial, etc.).

3.1.5 Negotiation

Cryptographic primitives are ordinarily used only by the implementors of cryptographic services. Negotiation should be used to establish which cryptographic service(s) are to be used, rather than to establish what primitives should be used. Ordinarily this negotiation will be done at a higher level than that of the cryptographic primitives and services themselves. No protocol for negotiating cryptographic primitives should be required.

3.2 Cryptographic Service Components

Figure 3-3 illustrates the Cryptographic Service Components:

Figure 3-3 Cryptographic Service Components

3.2.1 Function

These components provide access to cryptographic services such as data integrity and privacy protection ("data" here might be a file, a message, an i/o stream, etc...), key import and export, digital signature, keyed hash, etc....

Cryptographic Context Management provides the facilities through which applications initialize the cryptographic subsystem, activate keys for encryption and decryption, and clean up the state of the cryptographic subsystem after use.

Key usage controls permit control over a variety of aspects of key use, including how many times a key may be used; for what purposes it may be used (e.g. for signature only, for privacy only, for both signature and privacy, etc...), and so on.

Key derivation services permit generation of cryptographic-quality keys from non-key values such as passwords.

Crypto services are built on crypto primitives. A crypto service may support multiple implementations, each of which uses a different crypto primitive.

Descriptions of a few DES-based services will illustrate the difference between primitives and services; note that these are only examples:

- DEA is a crypto primitive which uses a 56-bit key and an initialization vector to transform a 64-bit plaintext into a 64-bit ciphertext.
- Data privacy is a crypto service. DES-CBC is an implementation of the cryptographic data privacy service which uses a 56-bit key, an initialization vector, and the DEA primitive to transform a plaintext of arbitrary length into a ciphertext of the same length subject to some rules defined by a "mode of operation". The rules describe how to "pad" plaintexts to a multiple of 64 bits and whether and how to induce dependencies among 64-bit blocks of the ciperhtext by feeding ciperhtext material from previous rounds of the encryption process into the current round.
- Data integrity is a crypto service. DES-CBC-MAC is an implementation of the data integrity service which uses the DEA primitive to generate a message authentication code given a 56-bit key, an initialization vector, and a plaintext of arbitrary length.

3.2.2 Protocols

Cryptographic services are typically called locally; it is not anticipated that any cryptographic service protocols will be standardized.

3.2.3 Interfaces

Candidate interfaces for cryptographic services include:

- Intel CSSM (CDSA)
- X/Open GCS-API
- Microsoft CryptoAPI 1.0
- SESAME CSF API

Other interfaces which may support some or all of the cryptographic primitive function include

- Cryptoki

- RSA BSAFE

Standardization of these interfaces would be of interest to developers of long-term-key service and protocol security service modules and to providers of cryptographic service modules. The APKI working group feels that it is important to standardize a single interface for cryptographic services, and recommends that the following interface be chosen as the basis for the standard:

- Intel CSSM

3.2.4 Profiles

Most cryptographic modules provide support for multiple services. Many crypto services are subject to legal restrictions on deployment (including both intellectual property encumbrances and national and international regulatory constraints on export, import, and deployment).

Cryptographic service profiles will have to be developed for PKI environments of interest (including, for example, the Internet, OMG CORBA, OSF DCE, Financial, etc.). These profiles will have to be developed with international deployment issues in mind. Each profile should be expressed in terms of the parameters used to select cryptographic services (and implementations of cryptographic services -- often called "mechanisms") through the cryptographic service interface (see the next section for more information on service and mechanism selection).

Profiles will need to specify, in addition to mechanism information, the data formats which each service can accept and return.

3.2.5 Negotiation

Negotiation of cryptographic services to be used by secure protocols and other security-aware applications is generally done at level higher than that of the cryptographic services themselves. The cryptographic service interface therefore must allow selection among available cryptographic services, and among available implementations of a single service, but it need not support negotiation.

3.3 Long-Term Key Services Components

Figure 3-4 illustrates the Long-Term Key Services Components; each component is described in more detail below.

3.3.1 Function

Key Lifecycle Management

The functions this component provides include key revocation, key repudiation, key expiration, and related services.

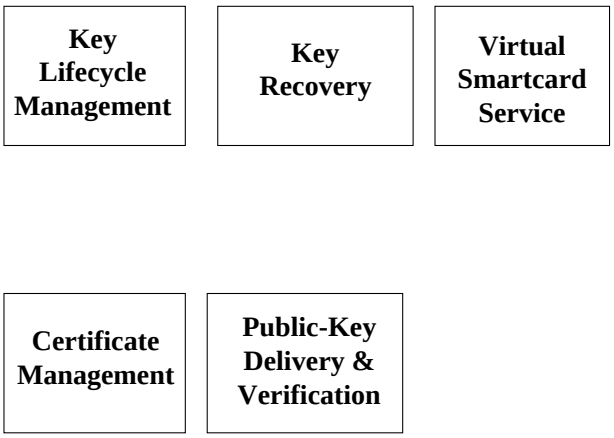
Key Recovery

This component supports preparation of keys for recovery, and permits later recovery under policy control.

Virtual Smartcard Service

The Virtual Smartcard Service Component permits users and other principals to store long-term personal security information (including private keys, certificates, and other

383

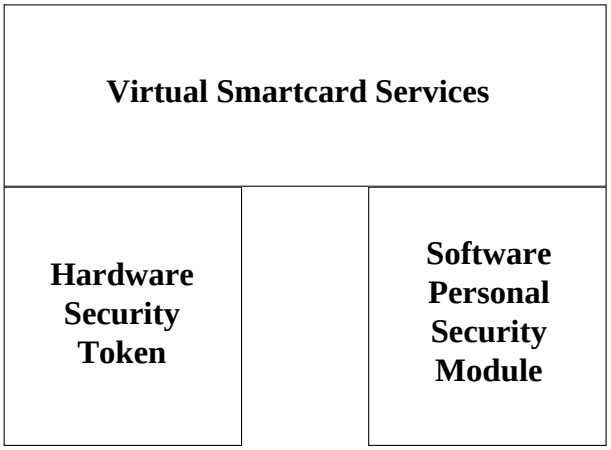


384 **Figure 3-4** Long Term Key Services Components

385 information) in protected storage, to activate personal keys for use via an authentication
386 procedure, and to use those keys for encryption, decryption, and signature activities.

387 Figure 3-5 illustrates the structure of this component.

388



389 **Figure 3-5** Virtual Smartcard Service Structure

390 **Certificate Management**

391 The Certificate Management component allows users, administrators and other principals
392 to request certification of public keys and revocation of previously certified keys. It may

optionally generate key pairs and provide key-pair recovery services. There are four Certificate Management sub-components:

- The Local Registration Authority provides interfaces for requesting generation of key-pairs and corresponding certificates, requesting certification of existing public keys, and requesting revocation of existing certificates.
- The Certification Authority Agent (CA Agent) provides interfaces for certifying existing public keys, generating and returning key pairs and corresponding certificates, revoking existing certificates. The CA Agent implements these interfaces by using the services of a Certification Authority (CA).
- The Certification Authority certifies public keys (returning the generated certificate) and generates certificate revocation lists. In some configurations it will be "off-line".
- The Publication Authority provides interfaces through which CAs and CA Agents can place certificates and CRLs into public repositories or transmit them directly to requestors.

Public-Key Delivery and Verification

This component allows a program to retrieve any principal's certificate, verify its validity, and extract the principal's certified public key from the certificate.

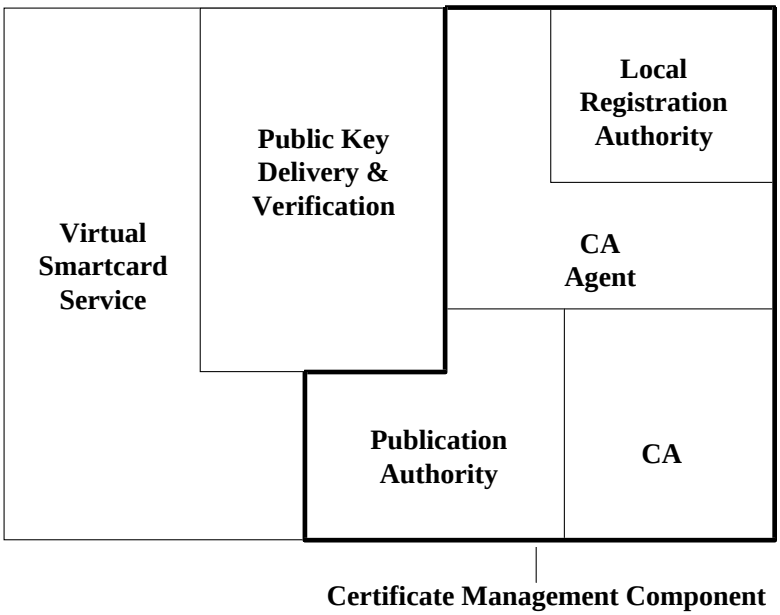


Figure 3-6 Public-Key Delivery and Verification Structures

Figure 3-6 illustrates the structure and interrelationships of the Certificate Management and Public-Key Delivery and Verification components and sub-components.

3.3.2 Protocols

Virtual Smartcard Service

When the Virtual Smartcard Service component is used for retrieval of user private keys, two models exist. One model (exemplified by PGP and Lotus Notes) manages private keys primarily on the client principal's machine (either in a software personal security module, or in a security token or other device external to the principal's workstation). In this model, no protocols are required for User/Principle Personal Security Info Management, since all operations are client-local.

The second model (exemplified by Novell NetWare) manages private keys at a central server and distributes them to client principals using a secure protocol. In this model, the client/server protocol for retrieval of private keys needs to be supported by the software personal security module subcomponent of the Virtual Smartcard Service component, as illustrated in Figure 3-7, (the dotted arrow in the figure represents the protocol):

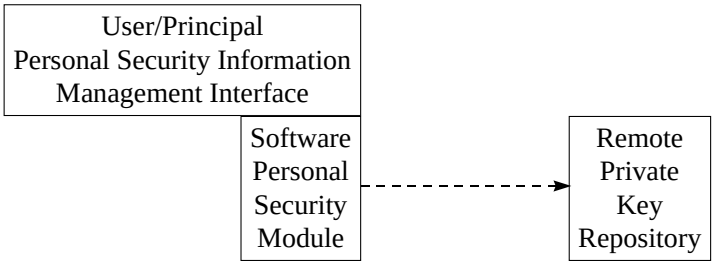


Figure 3-7 Virtual Smartcard Service Protocol

The APKI working group does not view standardization of this protocol to be essential.

Certificate Management

Protocols must be defined to permit creation, revocation, and update of certificates. Figure 3-8 illustrates Certificate Management protocols which might be standardized; each arrow in the diagram represents a protocol.

Note: Implementations may choose to assign the responsibility for generation of private keys (through use of the key generation facilities of the PKI architecture) to the CA, the LRA, or the User Workstation or Smartcard; additional protocols will be required to transmit the private key to the User Workstation or Smartcard if it is not generated there in the first place.

The APKI working group feels that the following protocols should be standardized at a minimum:

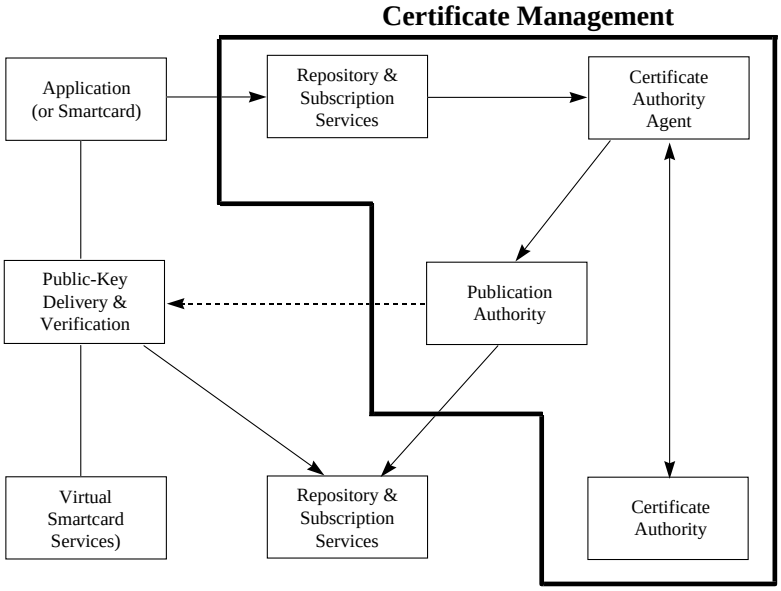


Figure 3-8 Certificate Management Protocols

- User Workstation or Smartcard to Certificate Management component
- Local Registration Authority to CA Agent

A candidate protocol specification including these protocols as well as a protocol for the Publication Authority to Public-Key Delivery protocol exists as IETF draft RFC ietf-pkix-ipki-part3-01.txt. The APKI working group endorses this proposal as the basis for standardization of the relevant APKI protocols.

Public-Key Delivery and Verification

Protocols must be defined to transport certificates and CRLs from the repositories in which they reside to the requester's machine. In the diagram, these protocols are represented by the arrows from the Publication Authority to the Public-Key Delivery and Verification component. The APKI working group feels that these protocols should be standardized. At least LDAP, email, and HTTP versions of these protocols should be defined.

A candidate protocol specification has been published as IETF draft RFC ietf-pkix-ipki2opp-00.txt. The APKI working group endorses this proposal as the basis for standardization of the relevant APKI protocols.

3.3.3 Interfaces

Virtual Smartcard Service

Candidate interfaces for this component include:

- PSM (HP Submission to OSF)
- SESAME CSF API

Other interfaces which may support some or all of the Virtual Smartcard Service functionality include:

- RSA PKCS-11
- PC Smartcard Consortium PC-SC specifications
- OpenCard framework
- Microsoft Wallet

The APKI working group feels that the Virtual Smartcard Service interface should be standardized.

Additionally, the APKI working group feels that the interface through which software communicates with Hardware Security Tokens should be standardized. A candidate interface for this functionality is:

- RSA PKCS-11

Public-Key Delivery and Verification

Candidate interfaces for this component include:

- SESAME PKM-API
- NSA CM-API
- Nortel CMS-API
- Intel CSSM (CDSA)

Other interfaces which may support some or all of the Public-Key Delivery and Verification function include

- Microsoft CryptoAPI version 2.0

The APKI working group feels that the Public-Key Delivery and Verification interface should be standardized. The APKI working group endorses the Intel CSSM interface, with extended Certificate and Key Lifecycle functionality currently being defined by The Open Group, as the base document for this interface standard.

Certificate Management

Candidate interfaces for this component include:

- Nortel CMS-API
- SESAME PKM API
- OSF RFC 80 API
- Intel CDSA

Other interfaces which may support some or all of the Certificate Management function include

- Microsoft CryptoAPI version 2.0

The APKI working group feels that the following interfaces should be standardized at a minimum:

- CA Agent
- Local Registration Authority

The APKI working group endorses the Intel CSSM interface, with extended Certificate and Key Lifecycle functionality currently being defined by The Open Group, as the base document for this interface standard.

Specification of the Publication Authority interface would also be useful to providers of repositories and communications protocols who wish to make their products available as certificate and CRL transmission media; a standard Publication Authority interface would allow them to provide Publication Authority services without requiring changes to CA Agent code.

3.3.4 Profiles

It is anticipated that multiple CAs will exist in typical PKI environments; individual servers may require the use of certificates with specific properties (signing CA, supported extensions, name format, etc...) Profiles for certificate format, contents, extensions, and policy will be needed for PKI environments of interest, including the Internet, Financial Industry, Credit Card Industry (for use with SET), Government, and Healthcare Industry environments.

A draft profile (for the Internet PKI environment) for certificate format, contents, and extensions exists as IETF draft RFC ietf-pkix-ipki-part1-01.txt. A draft policy profile for the Internet PKI environment has been published as IETF draft RFC ietf-pkix-ipki-part4-00.txt.

3.3.5 Negotiation

It is not anticipated that any of the Long-Term Key Services components will require negotiation protocols. The Certificate Management interfaces will need to provide a mechanism through which callers can identify which CA should issue certificates and CRLs requested through its interface, in case more than one CA is available.

The Virtual Smartcard Service interface will need to support selection of user/principal certificates for environments in which users have more than one certificate.

3.4 Protocol Security Services Components

Protocol security services are divided into two fundamental classes:

- Session-Oriented: security services which require exploiting entities to maintain security state information associated with protocol exchanges.
- Store & Forward: security services which encapsulate all required security state information inside the protected message tokens they generate; these services do not require exploiting entities to maintain security state information. Nonrepudiation services are necessarily store-and-forward services, because they must allow for "protection" of the nonrepudiability of a transaction after it has been completed and its state information destroyed. Nonrepudiation services are depicted separately from other store-and-forward protocol security services because, unlike store-and-forward data privacy and integrity services, use of Nonrepudiation services usually requires explicit user action.

Figure 3-9 illustrates the Protocol Security Services Components.

Figure 3-9 Protocol Security Services

3.4.1 Function

These components provide security services appropriate for use by designers of protocol stacks. Specifically, these components:

- Provide security mechanism and quality-of-protection negotiation protocols for use by communication partners needing to agree on a common security regime
- Manage security state information (if any) needed by protocol partners wishing to set up and maintain secure associations
- Encapsulate data origin authentication, data protection, and credential and privilege transport transparently within a single service (Crypto Services, by contrast, typically provide only data protection)
- Apply security mechanisms based on administered policy information

3.4.2 Protocols

Session-Oriented Protocol Security Services

A wide variety of protocol security services can be used to provide security for session-oriented protocols; examples which are described in existing or proposed Internet standards include the SPKM (which is Public-Key based), Kerberos (which is Secret-Key based), and SESAME (which has Public-Key, Secret-Key, and hybrid variants). Some of these services define their own protocols for run-time access to on-line security servers of a variety of types. All of them define formats for protected message tokens to be transported by their callers.

Store & Forward Protocol Security Services

Only a few protocol security services suitable for protection of store & forward protocol messages have been defined. The IDUP and SESAME services are proposed for Internet standardization. Both of these services define formats for protected message tokens to be transported by their callers.

Notary and Non-Repudiation Services.
These services must define formats for Non-Repudiation evidence tokens to be transmitted along with notarized data, and protocols implementing non-repudiable delivery and non-repudiable receipt.

The APKI working group feels that multiple protocol security services will continue to be required to meet the needs of diverse environments. No single standard for Session-Oriented, Store-and-Forward, or Nonrepudiation Protocol Security Services is proposed, therefore. The Protocol Security Services component interfaces will need to provide negotiation (for environments in which more than one service is available), and Protocol Security Service profiles will have to be established for PKI environments of interest.

3.4.3 Interfaces

The APKI working group feels that all of the Protocol Security Services interfaces should be standardized.

The structure of the Protocol Security Services is illustrated in Figure 3-10.

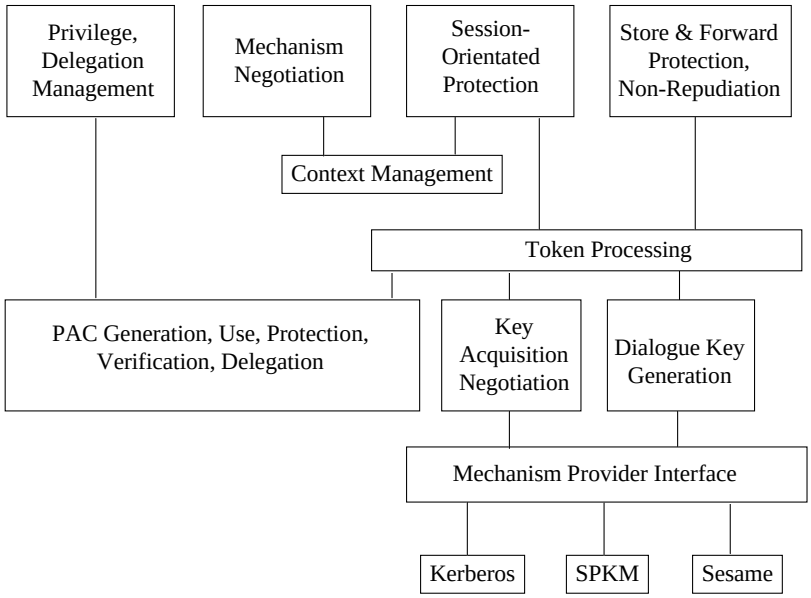


Figure 3-10 Protocol Security Service Structure

Session-Oriented Protocol Security Services
The preferred interface for these services is GSS-API (IETF RFC 1508).

Store & Forward Protocol Security Services
The preferred interface for these services is IDUP-GSS- API (IETF CAT draft ietf-cat-idup-gss-07.txt).

Non-Repudiation Services
The preferred interface for these services is IDUP-GSS- API (IETF CAT draft ietf-cat-idup-gss-07.txt).

In addition to these interfaces, the APKI working group feels that interfaces for Protection Mechanism Negotiation and Privilege and Delegation Management should be standardized.

The preferred interfaces for these services are draft-ietf-cat-gss-nego and draft-ietf-cat-xgss, respectively.

Other interfaces which may support some or all of the Protocol Security Services functionality include:

- Microsoft SSPI
- OMG CORBA Security
- TIPEM
- SHTTP

3.4.4 Profiles

GSS-API and IDUP-GSS-API are capable of supporting multiple security mechanisms; each API also allows selection of a wide range of qualities of data protection (e.g. strength of supported privacy protection, delegation mode, etc...) for each supported security mechanism.

Profiles will have to be developed to describe the set of preferred mechanisms and data protection quality parameters for PKI environments of interest. The APKI working group is not aware of a draft profile in this area.

3.4.5 Negotiation

Because they will be deployed in environments which require and provide multiple data protection mechanisms, the Protocol Security Services interfaces will need to support negotiation (of both protection mechanisms to be used and Quality of Protection to be applied).

A negotiation mechanism for GSS-API has been proposed and is described in IETF draft draft-ietf-cat-gss-snego-04.txt.

3.5 Secure Protocol Components

There are many kinds of secure protocols. Three important categories of secure protocols are:

- Connection-oriented peer-to-peer: These protocols allow exactly two partners, each of which must be on- line, to communicate securely.
- Connectionless peer-to-peer: These protocols allow exactly two partners, one or both of which may be off-line for some portion of the time interval during which messages are transmitted, to communicate securely.
- Connectionless multicast: These protocols allow one entity to communicate simultaneously and securely with several partners. Any or all entities may be off-line for some portion of the time interval during which messages are transmitted.

Figure 3-11 illustrates the Secure Protocol Components.

627

628 **Figure 3-11** Secure Protocol Components

629 3.5.1 Function

630 Secure protocols provide protected data transfer between communicating partners without
631 requiring any calls to security services. Applications using secure protocols may have to specify
632 a desired quality of protection before initiating a secure protocol exchange.

633 3.5.2 Protocols

634 Examples of secure protocols include:

635 • Connection-oriented peer-to-peer: Secure RPC, SSL, SHTTP, OMG SECIOP

636 • Connectionless peer-to-peer: IPSec, secure e-mail

637 • Connectionless multicast: Secure e-mail

638 3.5.3 Interfaces

639 Each secure protocol typically has its own interface.

3.5.4 Profiles

It is not yet clear whether profiles will be established for which Web transaction security protocols (e.g. SHTTP, HTTP-over-GSSAPI, etc...) should be used in which contexts.

3.5.5 Negotiation

The APKI working group feels that negotiation of secure protocols is outside the scope of the Public-Key (or even Security) Infrastructure effort.

3.6 System Security Enabling Components

Figure 3-12 illustrates the System Security Enabling Components.

Figure 3-12 System Security Enabling Components

3.6.1 Function

System functions (for example, Operating System functions) are needed to support user logon, user credential acquisition, and association of security state information with user processes and threads. For example, once a user has acquired credentials by authenticating himself to a smartcard, that user's processes should be able to use the smartcard interface to sign data using a private key stored on the smartcard. This will only be possible (and secure) if the system has maintained security state information associating the user's processes with the handle returned when the user authenticated himself to the smartcard.

It is not anticipated that the Internet Public-Key infrastructure will define any interfaces, protocols, profiles, or negotiation mechanisms in the area of System Security Enabling Services.

3.7 Security Policy Services Components

Figure 3-13 illustrates the Security Policy Service Components.

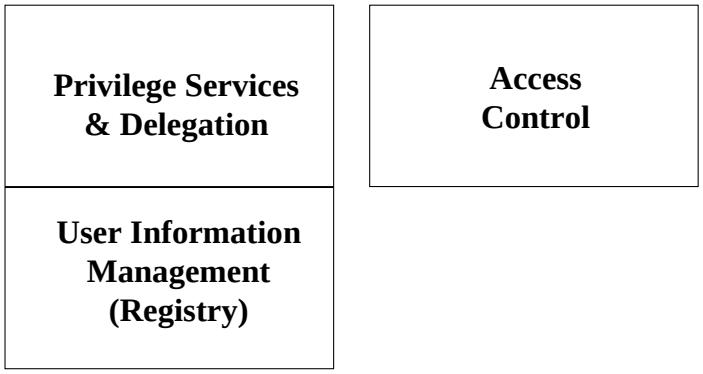


Figure 3-13 Security Policy Service Components

3.7.1 Function

Security Policy Services manage information about users' (and other principals') privileges and resource access control policies, and make access control decisions based on that information.

3.7.2 Protocols

Formats for privilege attribute tokens to be transported within secure protocols will need to be standardized. The most prominent existing privilege attribute format definitions today are those defined by ANSI X9, OSF DCE, SESAME, and the OMG CORBASEC standard. Privileges could be carried in X.509v3 certificate extensions, or in separate privilege attribute tokens.

3.7.3 Interfaces

It is not anticipated that the Internet Public-Key Infrastructure will define interfaces to privilege attribute services or access control services.

675 ### 3.7.4 Profiles

676 Interoperation of systems in differing security management domains will require
677 standardization of privilege attribute types and of the semantics of values of those types. No
678 proposed standard profile for privilege attributes exists today.

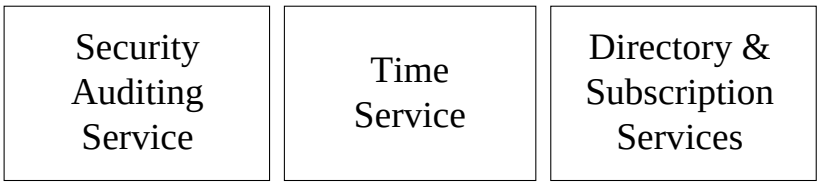
679 ### 3.7.5 Negotiation

680 <<TBD>>

681 ## 3.8 Supporting Services Components

682 Figure 3-14 lists the Supporting Services Components.

683



684 **Figure 3-14** Supporting Services Components

685 ### 3.8.1 Function

686 These components provide functions required by the security services or required for secure
687 operation of a networked system; however they do not enforce security policies.

3.8.2 Protocols

<<TBD>>

3.8.3 Interfaces

<<TBD>>

3.8.4 Profiles

<<TBD>>

3.8.5 Negotiation

<<Not germane to this document?>>

Chapter 4

Hardware Security Devices in the Architecture

The architecture is intended to support at least two kinds of hardware security devices:

Security Tokens

This class of devices includes smartcards, memory cards, time-synchronized tokens, and challenge- response tokens. These devices may provide crypto primitives and services, Virtual Smartcard services, and authentication functions.

Smartcards are assumed by the architecture to provide Virtual Smartcard Services. They will also frequently also provide at least the "Key Activation" and "Signing" components of Crypto Services; they may also provide other Crypto Services.

Memory cards provide only storage; Virtual Smartcard services involving state maintenance (e.g. key activation) or cryptography will have to be provided by the memory card's software drivers. Figure 4-1 illustrates how smartcards and memory cards can be used to support the Virtual Smartcard services.

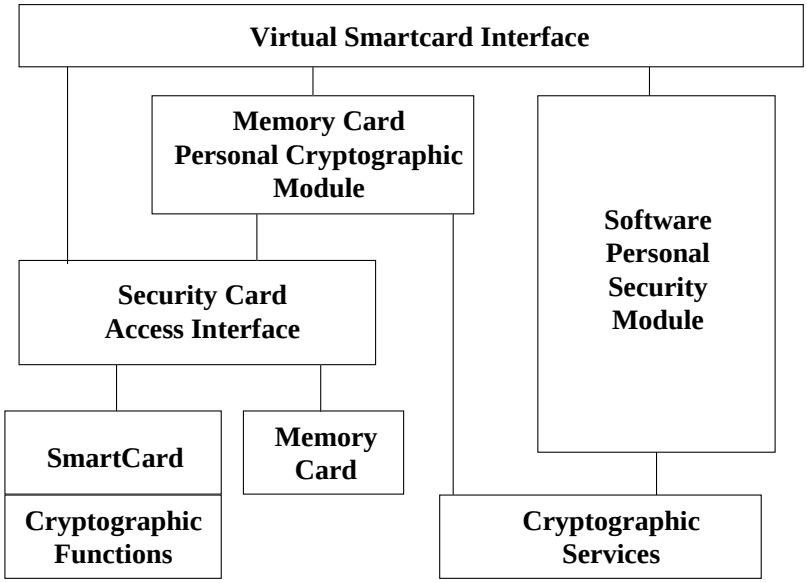


Figure 4-1 Hardware Security Devices

Time-synchronized and challenge-response tokens provide only authentication functionality, and will typically be integrated into the architecture through modifications to the System Security Enabling Services (particularly the "Logon" and "Obtain Credentials" components of those services).

Cryptographic Modules

This class of devices includes chipsets, bus-connected cryptographic adaptors, and remote cryptographic servers providing crypto primitives and services, but not providing user authentication functions.

Cryptographic modules are assumed by the architecture to provide the full range of Crypto Services (and they may provide direct access to some Crypto Primitives for the convenience of designers of new Crypto Services).

Glossary

Notes to Reviewers

This section with side shading will not appear in the final copy. - Ed.

This glossary has been extracted from the XDSF and requires editing to remove unwanted terms and to add SSO specific terms.

Note:

1. We use "confidentiality" and "privacy" interchangeably.
2. "Secret-key cryptography" is used to mean cryptography using a symmetric-key algorithm; "public-key" cryptography has the usual meaning; "private key" is used only to describe the private (secret) half of a key-pair generated for use with a public-key cryptographic system.

access control
The prevention of unauthorised use of a resource including the prevention of use of a resource in an unauthorised manner (see).

access control certificate
ADI in the form of a security certificate (see).

access control decision function
(ADF) — a specialised function that makes access control decisions by applying access control policy rules to a requested action, ACI (of initiators, targets, actions, or that retained from prior actions), and the context in which the request is made (see).

access control decision information
(ADI) —) the portion (possibly all) of the ACI made available to the ADF in making a particular access control decision (see).

access control enforcement function
(AEF) — a specialised function that is part of the access path between an initiator and a target on each access that enforces the decisions made by the ADF (see).

access control information
(ACI) — any information used for access control purposes, including contextual information (see).

access control list
A list of entities, together with their access rights which are authorised to have access to a resource (see).

access control policy
The set of rules that define the conditions under which an access may take place (see).

accountability
The property that ensures that the actions of an entity may be traced to that entity (see).

ACI
Access control information.

ACL
Access control list.

action
The operations and operands that form part of an attempted access (see).

action ADI
Action decision information associated with the action (see).

active threat
The threat of a deliberate unauthorised change to the state of the system

ADF
Access control decision function.

ADI
Access control decision information.

administrative security information
Persistent information associated with entities; it is conceptually stored in the Security Management Information Base. Examples are:

- security attributes associated with users and set up on user account installation, which is used to configure the user's identity and privileges within the system
- information configuring a secure interaction policy between one entity and another entity, which is used as the basis for the establishment of operational associations between those two entities.

AEF
Access control enforcement function.

alarm collector function
A function that collects the security alarm messages, translates them into security alarm records, and writes them to the security alarm log (see).

alarm examiner function
A function that interfaces with a security alarm administrator (see).

API
Application Programming Interface.

The interface between the application software and the application platform, across which all services are provided.

The application programming interface is primarily in support of application portability, but system and application interoperability are also supported by a communication API (see **Procurement Guide**).

assertion
Explicit statement in a system security policy that security measures in one security domain constitute an adequate basis for security measures (or lack of them) in another (see).

association-security-state
The collection of information that is relevant to the control of communications security for a particular application-association (see).

audit
See Security Audit (see).

audit authority
The manager responsible for defining those aspects of a security policy applicable to maintaining a security audit (see).

audit event detector function
A function that detects the occurrence of security-relevant events. This function is normally an inherent part of the functionality implementing the event (see).

audit recorder function
A function that records the security-relevant messages in a security audit trail (see).

audit trail
See Security Audit Trail (see).

audit trail analyser function
A function that checks a security audit trail in order to produce, if appropriate, security alarm messages (see).

audit trail archiver function
A function that archives a part of the security audit trail (see).

audit trail collector function
A function that collects individual audit trail records into a security audit trail (see).

audit trail examiner function
A function that builds security reports out of one or more security audit trails (see).

audit trail provider function
A function that provides security audit trails according to some criteria (see).

authenticated identity
An identity of a principal that has been assured through authentication (see).

authentication
Verify claimed identity; see data origin authentication, and peer entity authentication (see).

authentication certificate
Authentication information in the form of a security certificate which may be used to assure the identity of an entity guaranteed by an authentication authority (see).

authentication exchange
A sequence of one or more transfers of exchange authentication information (AI) for the purposes of performing an authentication (see).

authentication information (AI)
Information used to establish the validity of a claimed identity (see).

authentication initiator
The entity which starts an authentication exchange (see).

authentication method
Method for demonstrating knowledge of a secret. The quality of the authentication method, its strength is determined by the cryptographic basis of the key

distribution service on which it is based. A symmetric key based method, in which both entities share common authentication information, is considered to be a weaker method than an asymmetric key based method, in which not all the authentication information is shared by both entities.

authorisation
The granting of rights, which includes the granting of access based on access rights (see).

authorisation 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 authorisation policy may be defined in terms of access control lists, capabilities or attributes assigned to security subjects, security objects or both (see).

availability
The property of being accessible and usable upon demand by an authorised entity (see).

capability
A token used as an identifier for a resource such that possession of the token confers access rights for the resource (see).

ciphertext
Data produced through the use of encipherment. The semantic content of the resulting data is not available (see).

Note: Ciphertext may itself be input to encipherment, such that super-enciphered output is produced.

claim authentication information
(Claim AI) — information used by a claimant to generate exchange AI needed to authenticate a principal (see).

claimant
An entity which is or represents a principal for the purposes of authentication. A claimant includes the functions necessary for engaging in authentication exchanges on behalf of a principal (see).

clear text
Intelligible data, the semantic content of which is available (see).

client-server
These operations occur between a pair of communicating independent peer processes. The peer process initiating a service request is termed the client. The peer process responding to a service request is termed the server. A process may act as both client and server in the context of a set of transactions.

confidentiality
The property that information is not made available or disclosed to unauthorised individuals, entities, or processes (see).

contextual information
Information derived from the context in which an access is made (for example, time of day) (see).

corporate security policy
The set of laws, rules and practices that regulate how assets including sensitive information are managed, protected and distributed within a user organisation

(see).

countermeasure
The deployment of a set of security services to protect against a security threat.

credentials
Data that is transferred to establish the claimed identity of an entity (see).

cryptanalysis
The analysis of a cryptographic system and its inputs and outputs to derive confidential variables and/or sensitive data including clear text (see).

cryptographic algorithm
A method of performing a cryptographic transformation (see cryptography) on a data unit. Cryptographic algorithms may be based on symmetric key methods (the same key is used for both encipher and decipher transformations) or on asymmetric keys (different keys are used for encipher and decipher transformations).

cryptographic checkvalue
Information that is derived by performing a cryptographic transformation (see cryptography) on a data unit (see).

Note: The derivation of the checkvalue may be performed in one or more steps and is a result of a mathematical function of the key and data unit. It is usually used to check the integrity of a data unit.

cryptography
The discipline that embodies principles, means, and the methods for the transformation of data in order to hide its information content, prevent its undetected modification and/or prevent its unauthorised use (see).

Note: The choice of cryptography mechanism determines the methods used in encipherment and decipherment. An attack on a cryptographic principle, means or methods is cryptanalysis.

data integrity
The property that data has not been altered or destroyed in an unauthorised manner (see).

data origin authentication
The corroboration that the entity responsible for the creation of a set of data is the one claimed.

decipherment
The reversal of a corresponding reversible encipherment (see).

decryption
See decipherment (see).

denial of service
The unauthorised prevention of authorised access to resources or the delaying of time-critical operations (see).

digital fingerprint
A characteristic of a data item, such as a cryptographic checkvalue or the result of performing a one-way hash function on the data, that is sufficiently peculiar to the data item that it is computationally infeasible to find another data item that possesses the same characteristics (see).

digital signature
Data appended to, or a cryptographic transformation (see cryptography) of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery for example, by the recipient (see).

discretionary access control
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).

distinguishing identifier
Data that unambiguously distinguishes an entity in the authentication process. Such an identifier shall be unambiguous at least within a security domain (see).

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 (see).

encapsulated subsystem
A collection of procedures and data objects that is protected in a domain of its own so that the internal structure of a data object is accessible only to the procedures of the encapsulated subsystem and that those procedures may be called only at designated domain entry points. Encapsulated subsystem, protected subsystem and protected mechanisms of the TCB are terms that may be used interchangeably (see).

encipherment
The cryptographic transformation of data (see cryptography) to produce ciphertext (see).

Note: Encipherment may be irreversible, in which case the corresponding decipherment process cannot feasibly be performed. Such encipherment may be called a one-way-function or cryptochecksum.

encryption
See encipherment (see).

end-to-end encipherment
Encipherment of data within or at the source end system, with the corresponding decipherment occurring only within or at the destination end system (see).

exchange authentication information
(Exchange AI) — information exchanged between a claimant and a verifier during the process of authenticating a principal (see).

identification
The assignment of a name by which an entity can be referenced. The entity may be high level (such as a user) or low level (such as a process or communication channel.

identity-based security policy
A security policy based on the identities or attributes of users, a group of users, or entities acting on behalf of the users and the resources or targets being accessed (see).

initiator
An entity (for example, human user or computer based entity) that attempts to access other entities (see).

initiator access control decision information
(Initiator ADI) — ADI associated with the initiator (see).

initiator access control information
(Initiator ACI) — access control information relating to the initiator (see).

integrity
See Data Integrity (see).

key
A sequence of symbols that controls the operations of encipherment and decipherment (see).

key management
The generation, storage, distribution, deletion, archiving and application of keys in accordance with a security policy (see).

masquerade
The unauthorised pretence by an entity to be a different entity (see).

messaging application
An application based on a store and forward paradigm; it requires an appropriate security context to be bound with the message itself.

non-discretionary access control
A non-discretionary authorisation scheme is one under which only the recognised security authority of the security domain may assign or modify the ACI for the authorisation scheme such that the authorisations of principals under the scheme are modified.

off-line authentication certificate
A particular form of authentication information binding an entity to a cryptographic key, certified by a trusted authority, which may be used for authentication without directly interacting with the authority (see).

on-line authentication certificate
A particular form of authentication information, certified by a trusted authority, which may be used for authentication following direct interaction with the authority (see).

operational security information
Transient information related to a single operation or set of operations within the context of an operational association, for example, a user session. Operational security information represents the current security context of the operations and may be passed as parameters to the operational primitives or retrieved from the operations environment as defaults.

organisational security policy
Set of laws, rules, and practices that regulates how an organisation manages, protects, and distributes sensitive information (see).

password
Confidential authentication information, usually composed of a string of characters (see).

peer-entity authentication
The corroboration that a peer entity in an association is the one claimed (see).

physical security
The measures used to provide physical protection of resources against deliberate and accidental threats (see).

platform domain
A security domain encompassing the operating system, the entities and operations it supports and its security policy.

policy
See security policy (see).

primary service
An independent category of service such as operating system services, communication services and data management services. Each primary service provides a discrete set of functionality. Each primary service inherently includes generic qualities such as usability, manageability and security.

Security services are therefore not primary services but are invoked as part of the provision of primary services by the primary service provider.

principal
An entity whose identity can be authenticated (see).

privacy
The right of individuals to control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

Note: because this term relates to the right of individuals, it cannot be very precise and its use should be avoided except as a motivation for requiring security (see).

private key
A key used in an asymmetric algorithm. Possession of this key is restricted, usually to only one entity (see).

public key
The key, used in an asymmetric algorithm, that is publicly available (see).

quality of protection
A label that implies methods of security protection under a security policy. This normally includes a combination of integrity and confidentiality requirements and is typically implemented in a communications environment by a combination of cryptographic mechanisms.

repudiation
Denial by one of the entities involved in a communication of having participated in all or part of the communication (see).

rule-based security policy
A security policy based on global rules imposed for all users. These rules usually rely on a comparison of the sensitivity of the resources being accessed and the possession of corresponding attributes of users, a group of users, or entities acting on behalf of users (see).

seal
A cryptographic checkvalue that supports integrity but does not protect against forgery by the recipient (that is, it does not support non-repudiation). When a seal is associated with a data element, that data element is *sealed* (see).

secondary discretionary disclosure
An example of the misuse of access rights. It occurs when a principal authorised to access some information copies that information and authorises access to the copy by a second principal who is not authorised to access the original information.

secret key
In a symmetric cryptographic algorithm the key shared between two entities (see).

secure association
An instance of secure communication (using communication in the broad sense of space and/or time) which makes use of a secure context.

secure context
The existence of the necessary information for the correct operation of the security mechanisms at the appropriate place and time.

secure interaction policy
The common aspects of the security policies in effect at each of the communicating application processes (see).

security architecture
A high level description of the structure of a system, with security functions assigned to components within this structure (see).

security attribute
A security attribute is a piece of security information which is associated with an entity.

security audit
An independent review and examination of system records and operations in order to test for adequacy of system controls, to ensure compliance with established policy and operational procedures, to detect breaches in security and to recommend any indicated changes in control, policy and procedures (see).

security audit message
A message generated following the occurrence of an auditable security-related event (see).

security audit record
A single record in a security audit trail corresponding to a single security-related event (see).

security audit trail
Data collected and potentially used to facilitate a security audit (see).

security auditor
An individual or a process allowed to have access to the security audit trail and to build audit reports (see).

security aware
The caller of an API that is aware of the security functionality and parameters which may be provided by an API.

security certificate
A set of security-relevant data from an issuing security authority that is protected by integrity and data origin authentication, and includes an indication of a time period of validity (see).

Note: All certificates are deemed to be security certificates (see the relevant definitions in) adopted in order to avoid terminology conflicts with (that is the directory authentication standard).

security domain
A set of elements, a security policy, a security authority and a set of security-relevant operations in which the set of elements are subject to the security policy, administered by the security authority, for the specified operations (see).

security event manager
An individual or process allowed to specify and manage the events which may generate a security message and to establish the action or actions to be taken for each security message type (see).

security label
The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource (see).

Note: The marking may be explicit or implicit.

security policy
The set of criteria for the provision of security services (see also identity-based and rule-based security policy).

security service
A service which may be invoked directly or indirectly by functions within a system that ensures adequate security of the system or of data transfers between components of the system or with other systems.

security state
State information that is held in an open system and which is required for the provision of security services.

security token
A set of security-relevant data that is protected by integrity and data origin authentication from a source that is not considered a security authority (see).

security unaware
The caller of an API that is unaware of the security functionality and parameters which may be provided by an API.

sensitivity
The characteristic of a resource that implies its value or importance, and may include its vulnerability (see).

separation
The concept of keeping information of different security classes apart in a system (see).

Note: Separation may be implemented by temporal, physical, logical or cryptographic techniques.

service domain
A security domain encompassing an application, the entities and operations it supports and its security policy.

signature
See digital signature (see).

strength of mechanism
An aspect of the assessment of the effectiveness of a security mechanism, namely the ability of the security mechanism to withstand direct attack against deficiencies in its underlying algorithms, principles and properties (see).

system security function
A capability of an open system to perform security-related processing (see).

target
An entity to which access may be attempted (see).

target ADI
ADI associated with the target (see).

target ACI
Access control information relating to the target (see).

threat
A potential violation of security (see).
An action or event that might prejudice security (see).

traffic analysis
The inference of information from observation of traffic flows (presence, absence, amount, direction and frequency) (see).

traffic flow confidentiality
A confidentiality service to protect against traffic analysis (see).

traffic padding
The generation of spurious instances of communication, spurious data units or spurious data within data units (see).

trap door
A hidden software or hardware mechanism that permits system protection mechanisms to be circumvented. It is activated in some non-apparent manner (for example, special ‘‘random’’ key sequence at a terminal) (see).

trojan horse
Computer program containing an apparent or actual useful function that contains additional (hidden) functions that allow unauthorised collection, falsification or destruction of data (see).

trust
A relationship between two elements, a set of operations and a security policy in which element X trusts element Y if and only if X has confidence that Y behaves in a well defined way (with respect to the operations) that does not violate the given security policy (see).

trusted computing base (TCB)
The totality of protection mechanisms within an IT system, including hardware, firmware, software and data, the combination of which is responsible for enforcing the security policy.

trusted functionality
That which is perceived to be correct with respect to some criteria, for example, as established by a security policy (see).

trusted path
Mechanism by which a person using a terminal can communicate directly with the TCB (see).

Note: Trusted path can only be activated by the person or the TCB and cannot be imitated by untrusted software.

trusted third party
A security authority or its agent, trusted by other entities with respect to security-related operations (see).

verification AI
Information used by a verifier to verify an identity claimed through exchange AI (see).

verifier
An entity which is or represents the entity requiring an authenticated identity. A verifier includes the functions necessary for engaging in authentication exchanges (see).

virus
Self replicating, malicious program segment that attaches itself to an application or other executable system component and leaves no external signs of its presence (see).

vulnerability
Weakness in an information system or components (for example, system security procedures, hardware design, internal controls) that could be exploited to produce an information-related misfortune (see).

Index