Open Group Guide

Architecture for Public-Key Infrastructure (APKI)

The Open Group

© March 1999, The Open Group

All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system, or transmitted, in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the copyright owners.

Open Group Guide

Architecture for Public-Key Infrastructure (APKI)

ISBN: 1-85912-221-3
Document Number: G801

Published in the U.K. by The Open Group, March 1999.

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 the leading vendor-neutral, international consortium for buyers and suppliers of technology. Its mission is to cause the development of a viable global information infrastructure that is ubiquitous, trusted, reliable, and as easy-to-use as the telephone. The essential functionality embedded in this infrastructure is what we term the *IT DialTone*. The Open Group creates an environment where all elements involved in technology development can cooperate to deliver less costly and more flexible IT solutions.

Formed in 1996 by the merger of the X/Open Company Ltd. (founded in 1984) and the Open Software Foundation (founded in 1988), 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 meet its new mission, as well as to assist user organizations, vendors, and suppliers in the development and implementation of products supporting the adoption and proliferation of systems which conform to standard specifications.

With more than 200 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 Open Brand, represented by the ''X'' Device, that designates vendor products which conform to Open Group Product Standards
- Promoting the benefits of the IT DialTone 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 trademark on behalf of the industry.

Development of Product Standards

This process includes the identification of requirements for open systems and, now, the IT DialTone, development of Technical Standards (formerly 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.

The ''X'' Device is used by vendors to demonstrate that their products conform to the relevant Product Standard. By use of the Open Brand they guarantee, through the Open Brand Trade Mark License 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 documentation, the main part of which is focused on development of Technical Standards 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:

- *Technical Standards* (formerly *CAE Specifications*)

  The Open Group Technical Standards form the basis for our Product Standards. These Standards are intended to be used widely within the industry for product development and procurement purposes.

  Anyone developing products that implement a Technical Standard can enjoy the benefits of a single, widely supported industry standard. Where appropriate, they can demonstrate product compliance through the Open Brand. Technical Standards are published as soon as they are developed, so enabling vendors to proceed with development of conformant products without delay.

- *CAE Specifications*

  CAE Specifications and Developers' Specifications published prior to January 1998 have the same status as Technical Standards (see above).

- *Preliminary Specifications*

  Preliminary Specifications have usually addressed 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 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 Technical Standard. While the intent is to progress Preliminary Specifications to corresponding Technical Standards, 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 Technical Standards, in which case the relevant Technology Specification is superseded by a Technical Standard.

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 Technical Standards or Preliminary 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.

Versions and Issues of Specifications

As with all *live* documents, Technical Standards and 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/corrigenda*.

Ordering Information

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

This Document

This document is a Guide (see above).

- Chapter 1 describes the requirements on a Public-Key Infrastructure (PKI).
- Chapter 2 presents the high-level structure of the PKI Architecture by grouping the Architecture's components into broad functional categories.
- Chapter 3:
  - — 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 PKI Architecture, an interface or protocol must:

    1. Be described by a publicly-available specification
    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.
- Appendix A discusses the requirements for and approaches to Virtual Smartcards.
- A glossary and index are provided.

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

Trademarks

Motif®, OSF/1®, UNIX®, and the ''X Device''® are registered trademarks and IT DialTone™ and The Open Group™ are trademarks of The Open Group in the U.S. and other countries.

The Open Group acknowledges that there may be other products that might be covered by trademark protection and advises the reader to verify them independently.

Acknowledgements

The Open Group gratefully acknowledges the work of the Open Group Security Program Group, in particular the PKI Task Group, in the development of this Guide, and also the following individuals:

Anne Anderson, Hewlett-Packard Company
Charles Blauner, JP Morgan & Co. Inc.
Belinda Fairthorne, Fujitsu-ICL
Warwick Ford
Robert Jueneman, Novell, Inc.
Ellen McDermott, Open Market
Howard Melman, formerly OSF
Denis Pinkas, Groupe Bull
Walt Tuvell, formerly OSF
John Wray, Compaq Computer Corporation

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

Amdahl Corporation
Barclays Bank plc
BCTEL
Bellcore
Boeing
Bull
Citicorp
Digital Equipment Corporation
Dynasoft
Electronic Data Systems
Fujitsu-ICL
GUIDE International
Harris Corporation
Hewlett-Packard Company
IBM Corporation
Information & Support Group
Jet Propulsion Laboratory
JP Morgan & Co. Inc.
Lockheed Martin
Mitre
NCR Corporation
NIST
Nortel
Pacific Gas & Electric
SCO
Shell International
Siemens
Sun Microsystems, Inc.
Sweden Post
Telecom Finland Ltd.
The Open Group
U.K. Ministry of Defence
U.S. Dept. of Defense/DISA
U.S. Dept. of Defense/NSA
Veritas Software Corporation

Referenced Documents

Open Group Documentation

Further details about these and other Open Group documents can be found at: *http://www.opengroup.org/publications*.

- CAE Specification, December 1995, Generic Security Service API (GSS-API) Base (ISBN: 1-85912-131-4, C441), published by The Open Group.
- Technical Standard, December 1997, Common Security: CDSA and CSSM (ISBN: 1-85912-194-2, C707), published by The Open Group.
- Preliminary Specification, January 1997, Distributed Audit Service (XDAS) (ISBN: 1-85912-139-X, P441), published by The Open Group.
- Guide, December 1994, Distributed Security Framework (ISBN: 1-85912-071-7, G410), published by The Open Group.
- TOG RFC 80.0: DCE 1.2 Certification API—Functional Specification, J.Wray, January 1995

IETF RFCs

These documents may be found at: *http://www.ietf.org/rfc*.

- IETF RFC 2025: The Simple Public-Key GSS-API Mechanism (SPKM), C.Adams, October 1996

  This document describes how to use the SPKM protocol under a GSS-API interface
- IETF RFC 2030: Simple Network Time Protocol (SNTP), Version 4, D.Mills, October 1996
- IETF RFC 2078: The Generic Security Service Application Program Interface (GSS-API), Version 2, J.Linn, January 1997

  This document describes the GSS-API, Version 2.0 interface, which provides integrity and privacy services for session-oriented messages.
- IETF RFC 2251: Lightweight Directory Access Protocol, Version 3 (LDAPv3), T.Howes, S.Kille, and M.Wahl, December 1997
- IETF RFC 2409: The Internet Key Exchange (IKE), D.Carrel and D.Hawkins, November 1998
- IETF RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile, W.Ford, R.Housley, W.Polk, and D.Solo, January 1999

  This document describes the profiles for use of X.509v3 certificates and Version 2 Certificate Revocation Lists (CRLs) by users in the Internet environment.
- IETF RFC 2478: The Simple and Protected GSS-API Negotiation Mechanism, E.Baize and D.Pinkas, December 1998

  This document describes a proposed mechanism negotiation preamble protocol for use by protocol partners wishing to use GSS-API to establish a secure association.
- IETF RFC 2479: Independent Data Unit Protection Generic Security Service Application Program Interface (IDUP-GSS-API), C.Adams, December 1998

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 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols, C.Adams and S.Farrell, March 1999

  This document specifies a new protocol specifically developed for the purpose of transporting messages like those specified in CMMF and CRMF among PKI elements.

- IETF RFC 2511: Internet X.509 Certificate Request Message Format, C.Adams, D.Kemp, M.Myers, and D.Solo, March 1999

  Certificate Request Message Format (CRMF) specifies a format used to convey a request for a certificate to a Certification Authority (CA) or Registration Authority (RA).

- IETF RFC 2527: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework, S.Chokhani and W.Ford, March 1999

  This document defines Certificate Policies and Certification Practice Statements (CPSs) and their inter-relationship. This document provides a framework to assist the writers of certificate policies or CPSs with their tasks.

IETF PKIX Documentation

The latest versions of these documents can be found at: *http://www.ietf.org/ids.by.wg/pkix.html*.

- Certificate Management Messages over CMS, B.Fox, X.Liu, M.Myers, and J.Weinstein

  Certificate Management Messages over CMS (CMC) defines the means by which PKI clients and servers may exchange PKI messages when using IETF S/MIME Cryptographic Message Syntax (CMS), Version 3 as a transaction envelope.

- Internet X.509 Public Key Infrastructure Certificate Management Message Formats, C.Adams and M.Myers

  PKI Certificate Management Message Formats (CMMF) defines message formats to be used between a PKI client and a PKI server or service. It complements standard protocols such as the separately specified Certificate Management Protocol (CMP) or CMC.

- Internet X.509 Public Key Infrastructure LDAPv2 Schema, S.Boeyen, T.Howes, and P.Richard

  This document defines a minimal scheme necessary to support the use of LDAPv2 for certificate and CRL retrieval and related functions for PKIX.

- Internet X.509 Public Key Infrastructure Online Certificate Status Protocol, C.Adams, R.Ankney, S.Galperin, A.Malpani, and M.Myers

  The Online Certificate Status Protocol (OCSP) enables applications to determine the state of an identified certificate in a more timely fashion than is possible with the Certificate Revocation List (CRL) mechanism.

- Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP, P.Hoffman and R.Housley

  This document describes the use of the File Transfer Protocol (FTP) and the Hyper-Text Transfer Protocol (HTTP) to obtain certificates and CRLs from PKI repositories.

- Internet X.509 Public Key Infrastructure Operational Protocols: LDAPv2, S.Boeyen, T.Howes, and P.Richard

  This document describes the use of LDAPv2 as a protocol for publishing and retrieving certificates and CRLs from a certificate repository.

Other IETF Documentation

The latest versions of these documents can be found at: *http://www.ietf.org/ids.by.wg/cat.html*.

- C LDAP Application Program Interface

  Refer to *http://www.ietf.org/html.charters/ldapext-charter.html* for the current status of this API.

- Generic Security Service API, Version 2: C Bindings (IETF CAT GSS-API, Version 2.0 C Bindings), J.Wray

  This document defines the C bindings for GSS-API, Version 2.0.

- IETF CAT XGSS-API

  This document extends the GSS-API to allow support of security attributes in addition to a single identity and to allow fine control of delegation.

  Refer to *http://www.ietf.org/html.charters/cat-charter.html* for the current status of this API.

- IETF S/MIME Cryptographic Message Syntax (CMS), Version 3

  Refer to *http://www.ietf.org/html.charters/smime-charter.html* for the current status of CMS.

- Independent Data Unit Protection Generic Security Service Application Program Interface: C Bindings (IDUP-GSS-API C Bindings)

  This document defines the C bindings for IETF RFC 2479.

IPsec Working Group Documentation

The IETF IPsec Working Group documents can be found at: *http://www.ietf.org/ids.by.wg/ipsec.html*.

Standards Documentation

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

- ISO/IEC 10181: 1996, Information Technology — Open Systems Interconnection — Security Frameworks for Open Systems:

  ISO/IEC 10181-1: 1996, Part 1: Overview
  ISO/IEC 10181-2: 1996, Part 2: Authentication Framework
  ISO/IEC 10181-3: 1996, Part 3: Access Control Framework

- ISO/IEC 10745: 1995, Information Technology — Open Systems Interconnection — Upper Layers Security Model.

- 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.

Chapter 1

Requirements on a Public-Key Infrastructure

1.1 Baseline Requirements for a Global PKI

An interoperable global Public-Key Infrastructure (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 Lifecycle Management

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

1. 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).

2. 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 (CA) to revoke certificates for individual keys under the terms of the applicable policy
   - Ability of a CA to suspend and reactivate certificates for individual keys under the terms of the applicable policy
   - Ability of a CA 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 (for example, 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 the owner of the retrieved key.

3. 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 ensure 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 the PKI and in data recovery centers.

  Note: Public and private key parts do not have the same lifecycle 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.

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 CA to another
5. CA 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 CA and repository functions in accordance with the governance policy—changes to certificate repositories must be transactional (for example, 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 APIs to and protocols between the component services of the PKI.

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

1.1.4 Recommendations

Adopt X.509v3 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. In particular, adopt the service interfaces defined in CDSA, Version 2.0 and the protocols defined in the PKIX series of IETF draft standards.

1.2 The Importance of Architecture

The PKI Task Group feels that a robust, flexible, standard, open PKI 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 PKI Architecture.

1.2.2 Protocols

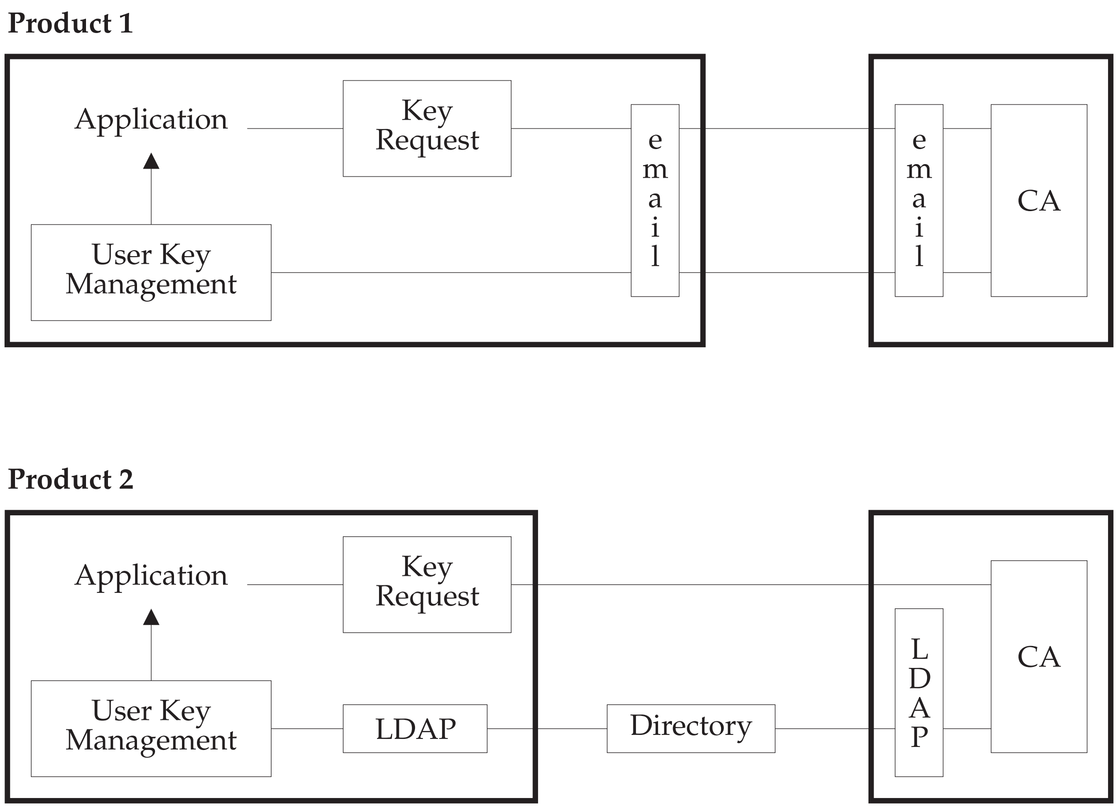

Figure 1-1 Protocols in Certificate Management

Figure 1-1 illustrates two Certificate Management products:

- Product 1 communicates key requests to the CA via electronic mail, and receives keys and certificates from the CA via electronic mail.
- 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.

1.2.3 Interfaces

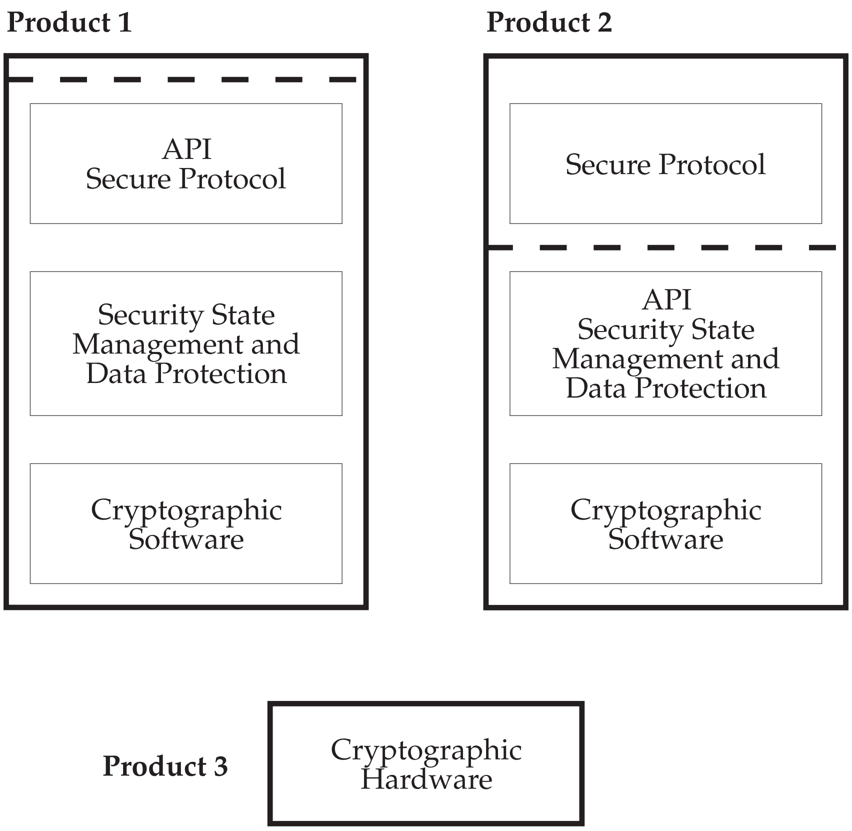

Figure 1-2 Example Security Products

Figure 1-2 illustrates a system on which three security products have been installed:

- 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.

The benefits of using standard interfaces include:

- Replaceability of services (for example, 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)

1.2.4 Profiles

Many of the services in the PKI Architecture can be implemented using a variety of different mechanisms and protocols (for example, 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

PKI Architecture and Summary of Recommendations

2.1 PKI Architecture Overview

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 their actions in the system.
- *Cryptographic Primitives and Services* provide the cryptographic functions on which public-key security is based (including secret-key primitives, such as the Data Encryption Standard (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, non-repudiation) 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:

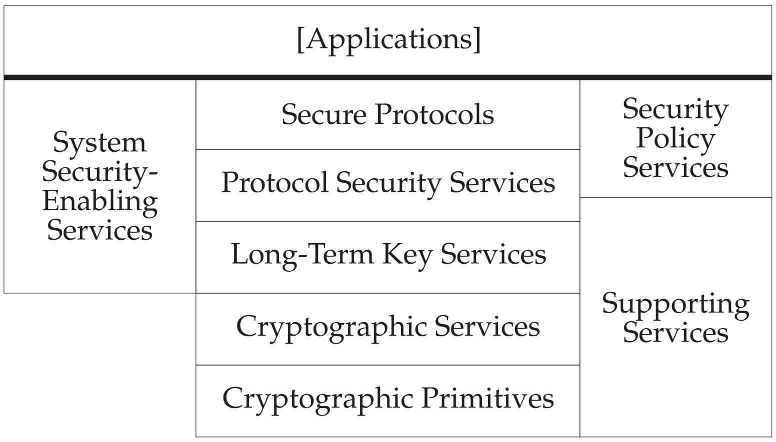

Figure 2-1 PKI Architecture Overview

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 ensure that keys, security context data, and policy data are appropriately protected in such environments.

2.2 Summary of Recommendations

2.2.1 System Security-Enabling Services

It is not anticipated that the Internet PKI will define any interfaces, protocols, profiles, or negotiation mechanisms in the area of system security-enabling services.

2.2.2 Cryptographic Primitives

Protocols — N/A.

Interfaces — Standardization of cryptographic primitive interfaces is not viewed as essential. However, the use of CSSM from CDSA, Version 2.0 is recommended.

Profiles — Profiles need to be developed for PKI environments of interest.

Negotiation — N/A.

2.2.3 Cryptographic Services

Protocols — N/A.

Interfaces — The PKI Task Group believes that it is important to standardize a single interface for cryptographic services, and recommends that CSSM from CDSA, Version 2.0 be chosen as the basis for the standard.

Profiles — Cryptographic service profiles will have to be developed for PKI environments of interest (including, for example, the Internet, OMG CORBA, OSF DCE, Financial, and so on). 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 Section 3.3.5 on page 21 for more information on service and mechanism selection).

Negotiation — N/A.

2.2.4 Long-Term Key Services

Protocols — The PKI Task Group endorses use of the IETF PKIX management protocols as the basis for standardization of the relevant PKI Architecture Certificate Management protocols. These IETF PKIX specifications are:

- IETF RFC 2511: Internet X.509 Certificate Request Message Format
- Internet X.509 Public Key Infrastructure Certificate Management Message Formats
- IETF RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols

- Certificate Management Messages over CMS

The PKI Task Group endorses use of the IETF PKIX operational protocols as the basis for standardization of the relevant PKI Architecture Public-Key Delivery and Verification protocols. These IETF PKIX specifications are:

- Internet X.509 Public Key Infrastructure Online Certificate Status Protocol
- Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP
- Internet X.509 Public Key Infrastructure Operational Protocols: LDAPv2
- Internet X.509 Public Key Infrastructure LDAPv2 Schema

Interfaces

The PKI Task Group recommends that the Key Lifecycle Management interface should be standardized and endorses the use of CSSM from CDSA, Version 2.0.

The PKI Task Group recommends that the Key Recovery interface should be standardized and endorses the use of the CSSM Key Recovery API from CDSA, Version 2.0.

The PKI Task Group recommends that the Public-Key Delivery and Verification interfaces should be standardized. The PKI Task Group endorses the CSSM API from CDSA, Version 2.0 as the base document for this interface standard.

The PKI Task Group recommends that the following Certificate Management interfaces should be standardized at a minimum:

- CA Agent
- Local Registration Authority

The PKI Task Group endorses the CSSM API from CDSA, Version 2.0 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.

Profiles

A profile (for the Internet PKI environment) for certificate format, contents, and extensions exists as IETF RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile.

A policy profile for the Internet PKI environment has been published as IETF RFC 2527: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework.

Negotiation

N/A.

2.2.5 Protocol Security Services

Protocols — Multiple protocols will continue to be required. Therefore no single standard is proposed in this area.

Interfaces

- The preferred interface for session-oriented protocol security services is IETF RFC 2078: The GSS-API, Version 2. (The C bindings for this specification are defined in Generic Security Service API, Version 2: C Bindings.)
- The preferred interface for store-and-forward protocol security Services is IETF RFC 2479: IDUP-GSS-API.
- The preferred interface for non-repudiation services is IETF RFC 2479: IDUP-GSS-API.

In addition to these interfaces, the PKI Task Group recommends that interfaces for protection mechanism negotiation and privilege and delegation management should be standardized. The preferred interfaces for these services are IETF RFC 2478: The Simple and Protected GSSI Negotiation Mechanism and IETF CAT XGSS-API, respectively.

Profiles — Profiles will need to be developed, but the PKI Task Group is not aware of any such profiles that have been defined.

Negotiation — A negotiation mechanism for GSS-API has been proposed and is described in IETF RFC 2478: The Simple and Protected GSSI Negotiation Mechanism.

2.2.6 Secure Protocols

Protocols — The IETF IPsec Working Group has produced work on secure protocols, notably IPsec (see **Referenced Documents** on page xiii) and IKE (see IETF RFC 2409: The Internet Key Exchange (IKE)).

Interfaces — Each secure protocol typically has its own interface.

Profiles — It is not yet clear whether profiles will be established for which Web transaction security protocols (for example, SHTTP, HTTPS, HTTP-over-GSS-API, and so on) should be used in which contexts.

Negotiation — The PKI Task Group considers that negotiation of secure protocols is outside the scope of the PKI (or even Security Infrastructure) effort.

2.2.7 Security Policy Services

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[1] standard. Privileges could be carried in X.509v3 certificate extensions, or in separate privilege attribute tokens.

Interfaces — It is not anticipated that the Internet PKI will define interfaces to privilege attribute services or access control services.

---

1. Further information about OMG and CORBA can be found at: *http://www.omg.org*.

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

Negotiation — N/A.

2.2.8 Supporting Services

Protocols — The Distributed Audit Service (XDAS) specification defines an API as well as a common audit record format that is applicable.

It is recommended that IETF RFC 2030: Simple Network Time Protocol (SNTP) is adopted as the time service protocol for use within the PKI Architecture.

It is recommended that IETF RFC 2251: Lightweight Directory Access Protocol, Version 3 (LDAPv3) is used as the basic directory service protocol within the PKI Architecture.

Interfaces — It is recommended that the XDAS Distributed Audit Service API is adopted as the security auditing service API.

Components of the PKI Architecture will access time via the interface provided by the supporting operating system.

It is recommended that the C LDAP Application Program Interface is used as the interface to directory services.

Profiles — Profiles are currently not defined in the areas of audit and time services. Profiles for directory-based services are being developed within both The Open Group and IETF based upon the LDAP protocol. See *http://www.opengroup.org/ldap*.

Negotiation — N/A.

Chapter 3

Public-Key Infrastructure Components

Figure 2-1 outlined the functional categories comprising the PKI Architecture and their relationships. It is repeated here as Figure 3-1:

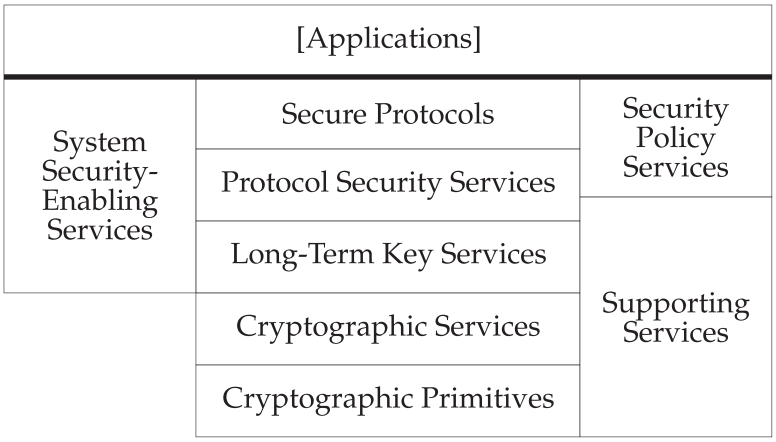

Figure 3-1 PKI Architecture

Each of the following sections describes one of the PKI Architecture's categories in detail, enumerating its components, and describing component functions, interfaces, and protocols.

Note: The scope of this current version of the PKI Architecture does not include object-oriented approaches such as JAVA. A subsequent version of this document will include object-oriented approaches where applicable.

3.1 System Security-Enabling Components

Figure 3-2 illustrates the system security-enabling components:

Figure 3-2 System Security-Enabling Components

3.1.1 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 PKI will define any interfaces, protocols, profiles, or negotiation mechanisms in the area of system security-enabling services.

3.2 Cryptographic Primitive Components

Figure 3-3 illustrates the cryptographic primitive components:

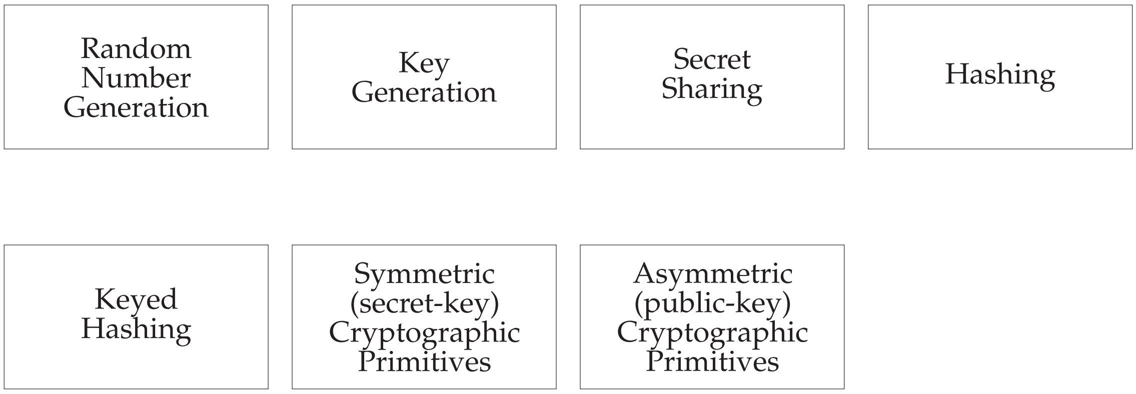

Figure 3-3 Cryptographic Primitive Components

Note: The PKI Architecture's cryptographic primitives may be provided by hardware (for example, Smartcards or cryptographic modules) or by software.

3.2.1 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, and so on.

3.2.2 Protocols

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

3.2.3 Interfaces

Candidate interfaces for access to cryptographic primitives include:

- The RSA BSafe library interface
- RSA PKCS-11
- CSSM API (from CDSA, Version 2.0)
- The Microsoft CryptoAPI 2.0

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 PKI Task Group, however, is that cryptographic functionality will ordinarily be used through the cryptographic service interfaces rather than through the cryptographic primitive interfaces.

Recommendation: Standardization of cryptographic primitive interfaces is not viewed as essential. However, the PKI Task Group recommends the use of CSSM from CDSA, Version 2.0.

Note: The Common Security Services Manager (CSSM) is the core of CDSA. CSSM manages categories of security services and multiple discrete implementations of those services as add-in security modules. CSSM:

- Defines the API for accessing security services
- Defines the service provider's interface for security service modules
- Dynamically extends the security services available to an application, while maintaining an extended security perimeter for that application

Applications request security services through the CSSM Security API, or layered security services and tools implemented over the CSSM API. The requested security services are performed by add-in security modules. Four basic types of module managers are defined:

- Cryptographic Services Manager
- Trust Policy Services Manager
- Certificate Library Services Manager
- Data Storage Library Services Manager

CSSM supports elective module managers that dynamically extend the system with new categories of security services providing for future extensibility. See CDSA, Version 2.0 for further details.

3.2.4 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, and so on).

3.2.5 Negotiation

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

3.3 Cryptographic Service Components

Figure 3-4 illustrates the cryptographic service components:

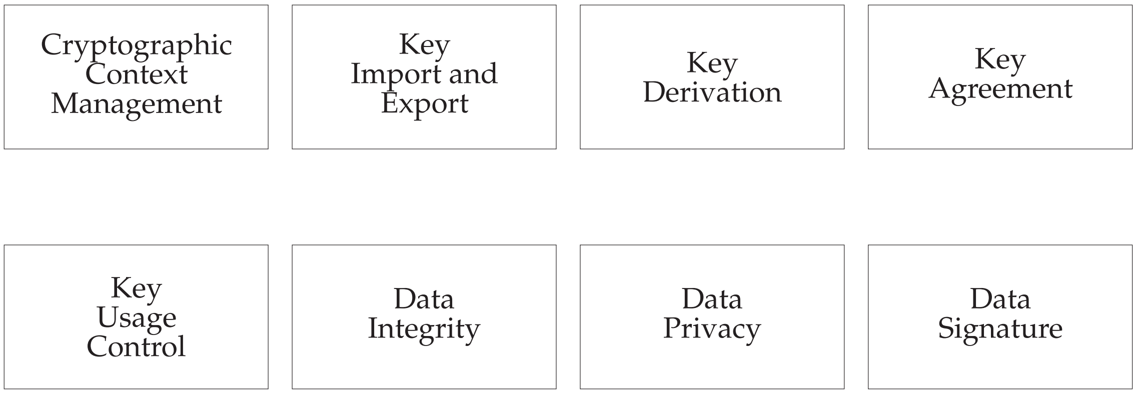

Figure 3-4 Cryptographic Service Components

3.3.1 Function

These components provide access to cryptographic services such as data integrity and confidentiality protection (''data'' here might be a file, a message, an I/O stream, and so on), key import and export, digital signature, keyed hash, and so on.

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 (for example, for signature only, for confidentiality only, for both signature and confidentiality), and so on.

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

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

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

- The Data Encryption Algorithm (DEA) is a cryptographic primitive which uses a 56-bit key and an initialization vector to transform a 64-bit plaintext into a 64-bit ciphertext. It forms the basis of the DES.
- Data confidentiality is a cryptographic service. DES-CBC is an implementation of the cryptographic data confidentiality 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 ciphertext by feeding ciphertext material from previous rounds of the encryption process into the current round.
- Data integrity is a cryptographic 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.

3.3.2 Protocols

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

3.3.3 Interfaces

Candidate interfaces for cryptographic services include:

- CSSM API (from CDSA, Version 2.0)
- Microsoft CryptoAPI 2.0
- SESAME CSF API
- Advanced Encryption Standard (AES)

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.

Recommendation: The PKI Task Group believes that it is important to standardize a single interface for cryptographic services, and recommends that CSSM from CDSA, Version 2.0 be chosen as the basis for the standard.

3.3.4 Profiles

Most cryptographic modules provide support for multiple services. Many cryptographic 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, and so on). 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 Section 3.3.5 on page 21 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.

3.3.5 Negotiation

Negotiation of cryptographic services to be used by secure protocols and other security-aware applications is generally done at a 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.

3.4 Long-Term Key Services Components

Figure 3-5 illustrates the long-term key services components; each component is described in more detail below:

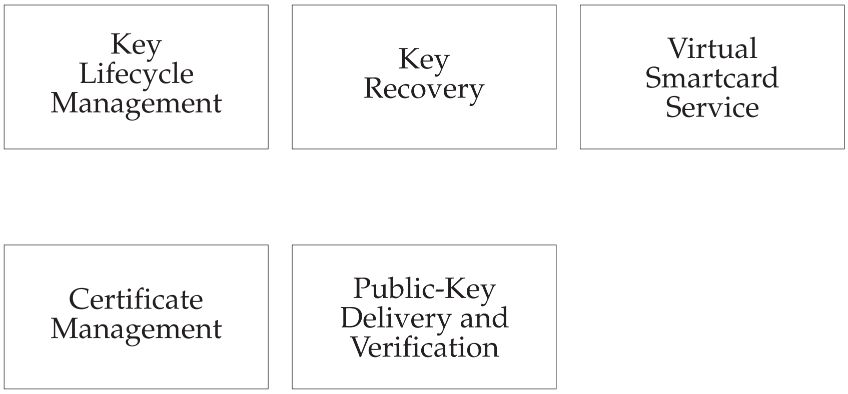

Figure 3-5 Long-Term Key Services Components

3.4.1 Function

Key Lifecycle Management

The functions this component provides include key generation, certificate request, 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 information) in protected storage, to activate personal keys for use via an authentication procedure, and to use those keys for encryption, decryption, and signature activities.

The Virtual Smartcard service is still subject to debate and therefore a detailed discussion of this service component is deferred to Appendix A.

Certificate Management

The Certificate Management component allows users, administrators, and other principals 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, and revoking

existing certificates. The CA Agent implements these interfaces by using the services of a Certification Authority (CA).

- The CA certifies public keys (returning the generated certificate) and generates Certificate Revocation Lists (CRLs). 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 requesters.

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.

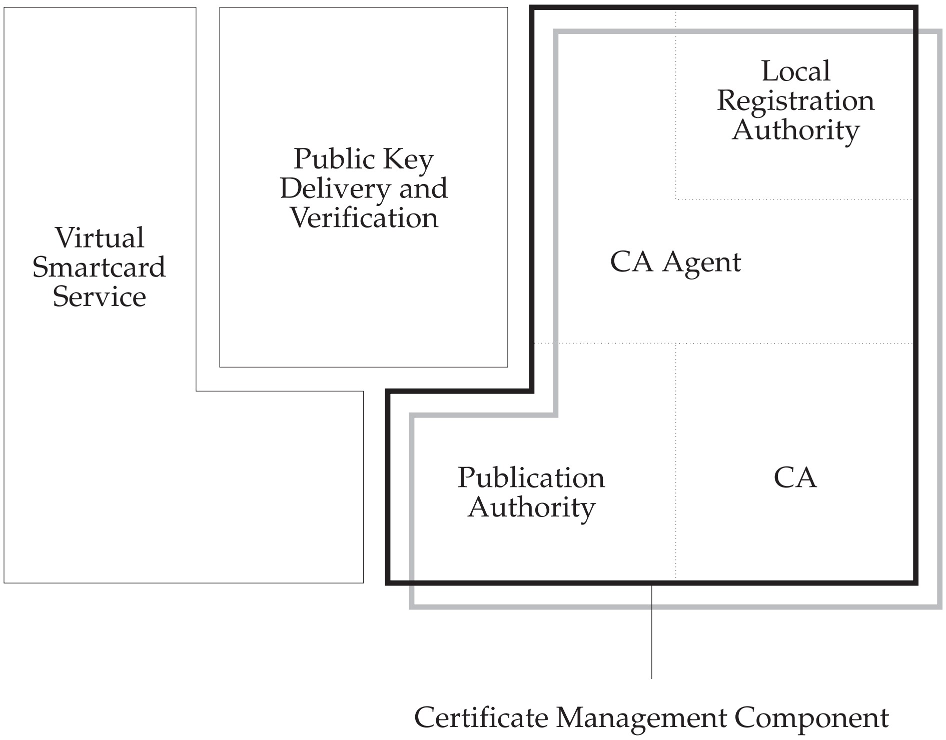

Figure 3-6 Public-Key Delivery and Verification Structures

Figure 3-6 is illustrative of the logical structure and inter-relationships of the Certificate Management and public-key delivery and verification components and sub-components.

3.4.2 Protocols

Key Lifecycle Management

Key Lifecycle Management is supported by the following IETF PKIX specifications:

- IETF RFC 2511: Internet X.509 Certificate Request Message Format
- Internet X.509 Public Key Infrastructure Certificate Management Message Formats
- IETF RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols
- Certificate Management Messages over CMS

Key Recovery

Key Recovery request and response formats are defined in:

- Internet X.509 Public Key Infrastructure Certificate Management Message Formats

Key request/response protocols are defined in:

- IETF RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols
- Certificate Management Messages over CMS

Virtual Smartcard Service

The Virtual Smartcard service is still subject to debate and therefore a detailed discussion of this service component is deferred to Appendix A on page 41.

Certificate Management

Protocols must be defined to permit creation, revocation, and refreshment of certificates. Figure 3-7 illustrates Certificate Management protocols which might be standardized; each arrow in the diagram represents a protocol.

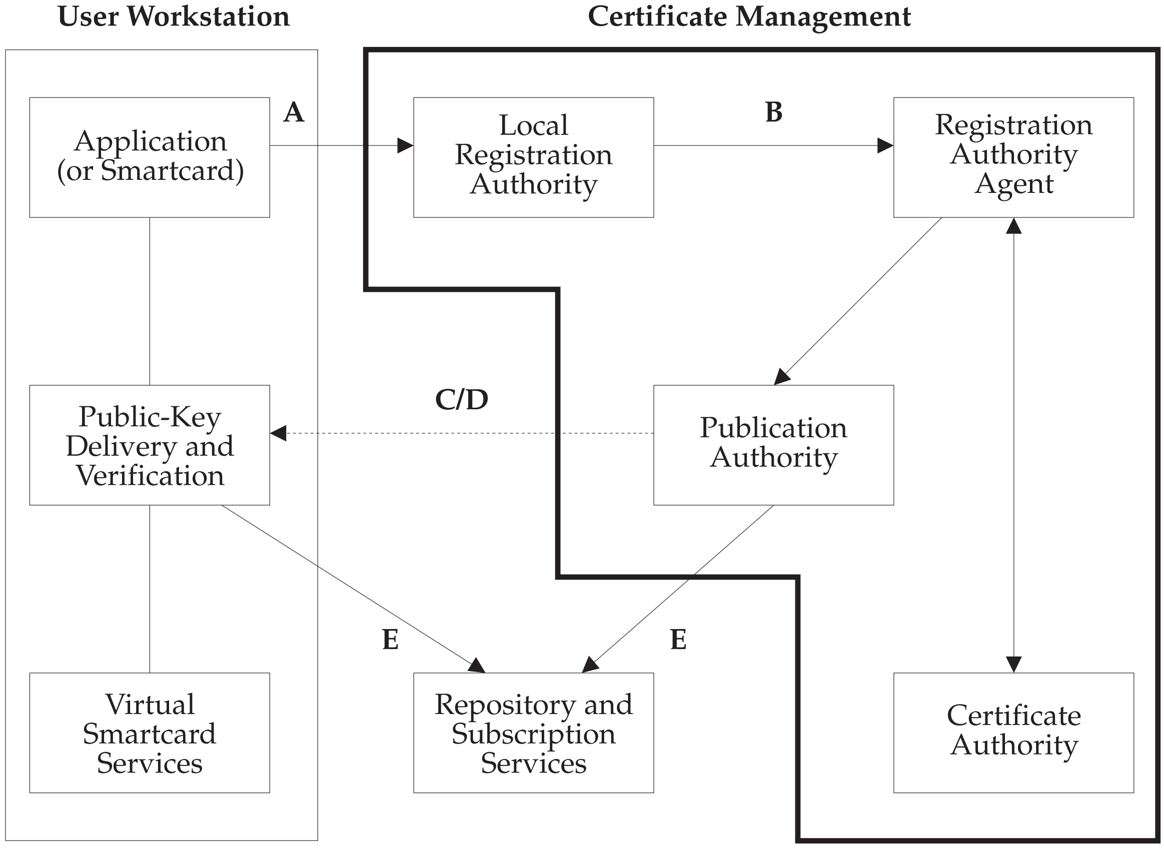

Figure 3-7 Certificate Management Protocols

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 Local Registration Authority, 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 PKI Task Group recommends that the following protocols should be standardized at a minimum:

- Protocol A: User Workstation or Smartcard to Certificate Management component
- Protocol B: Local Registration Authority to Registration Authority/Certificate Authority

Recommendation: In order to standardize the relevant PKI Architecture Certificate Management protocols, the PKI Task Group endorses the use of the following IETF PKIX specifications:

- IETF RFC 2511: Internet X.509 Certificate Request Message Format
- Internet X.509 Public Key Infrastructure Certificate Management Message Formats
- IETF RFC 2510: Internet X.509 Public Key Infrastructure Certificate Management Protocols
- Certificate Management Messages over CMS

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 (Protocols C, D, and E). The PKI Task Group recommends that these protocols should be standardized. At least LDAP, email, and HTTP versions of these protocols should be defined.

Candidate protocol draft specifications have been published as follows:

- LDAP-based protocol (Protocol E): Internet X.509 Public Key Infrastructure Operational Protocols: LDAPv2
- FTP/HTTP-based protocols (Protocols C and D): Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP

Recommendation: In order to standardize the relevant PKI Architecture Public Key Delivery and Verification protocols, the PKI Task Group endorses the use of the following IETF PKIX specifications:

- Internet X.509 Public Key Infrastructure Online Certificate Status Protocol
- Internet X.509 Public Key Infrastructure Operational Protocols: FTP and HTTP
- Internet X.509 Public Key Infrastructure Operational Protocols: LDAPv2
- Internet X.509 Public Key Infrastructure LDAPv2 Schema

Note: The IETF PKIX Working Group are defining a set of message formats and protocols for use within a PKI environment. At a high level, the set of operations for which management messages are being defined are:

- CA establishment
- End entity initialization
- Certification
- Certificate/CRL discovery operations
- Recovery operations
- Revocation operations
- Personal security environment operations

Readers are referred to the latest versions of the PKIX work accessible at *http://www.ietf.org/ids.by.wg/pkix.html*.

3.4.3 Interfaces

Key Lifecycle Management

Candidate interfaces for this component include:

- CSSM (from CDSA, Version 2.0)

Recommendation: The PKI Task Group recommends that the Key Lifecycle Management interface should be standardized and endorses the use of CSSM (from CDSA, Version 2.0).

Key Recovery

Candidate interfaces for this component include:

- CSSM Key Recovery API (from CDSA, Version 2.0)

Recommendation: The PKI Task Group recommends that the Key Recovery interface should be standardized and endorses the use of the CSSM Key Recovery API (from CDSA, Version 2.0).

Virtual Smartcard Service

The Virtual Smartcard service is still subject to debate and therefore a detailed discussion of this service component is deferred to Appendix A.

Public-Key Delivery and Verification

Candidate interfaces for this component include:

- CSSM API (from CDSA, Version 2.0)

Other interfaces which may support some or all of the public-key Delivery and Verification function include:

- Microsoft CryptoAPI, Version 2.0

Recommendation: The PKI Task Group recommends that the public-key Delivery and Verification interface should be standardized. The PKI Task Group endorses the CSSM API (from CDSA, Version 2.0), as the base document for this interface standard.

Certificate Management

Candidate interfaces for this component include:

- TOG RFC 80.0
- CSSM (from CDSA, Version 2.0)

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

- Microsoft CryptoAPI, Version 2.0

Recommendation: The PKI Task Group recommends that the following Certificate Management interfaces should be standardized at a minimum:

- CA Agent
- Local Registration Authority

The PKI Task group endorses the CSSM API (from CDSA, Version 2.0) 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.

3.4.4 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, and so on). 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 profile (for the Internet PKI environment) for certificate format, contents, and extensions exists as IETF RFC 2459: Internet X.509 Public Key Infrastructure Certificate and CRL Profile. A policy profile for the Internet PKI environment has been published as IETF RFC 2527: Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework.

3.4.5 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.

3.5 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-and-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. Non-repudiation services are necessarily store-and-forward services, because they must allow for ‘‘protection’’ of the non-repudiability of a transaction after it has been completed and its state information destroyed. Non-repudiation services are depicted separately from other store-and-forward protocol security services because, unlike store-and-forward data confidentiality and integrity services, use of non-repudiation services usually requires explicit user action.

Figure 3-8 illustrates the protocol security services components:

Figure 3-8 Protocol Security Services

3.5.1 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 (cryptographic services, by contrast, typically provide only data protection)
- Apply security mechanisms based on administered policy information

3.5.2 Protocols

The PKI Task Group believes that multiple protocol security services will continue to be required to meet the needs of diverse environments. Therefore, no single standard for session-oriented, store-and-forward, or non-repudiation protocol security services is proposed. 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.

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 SPKM (which is public-key based—see IETF RFC 2025: The Simple Public-Key GSS-API Mechanism (SPKM)), 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-and-Forward Protocol Security Services

Only a few protocol security services suitable for protection of store-and-forward protocol messages have been defined. The IDUP (see IETF RFC 2479: IDUP-GSS-API) 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.

3.5.3 Interfaces

Recommendation: The PKI Task Group recommends that all of the protocol security services interfaces should be standardized.

The structure of the protocol security services is illustrated in Figure 3-9:

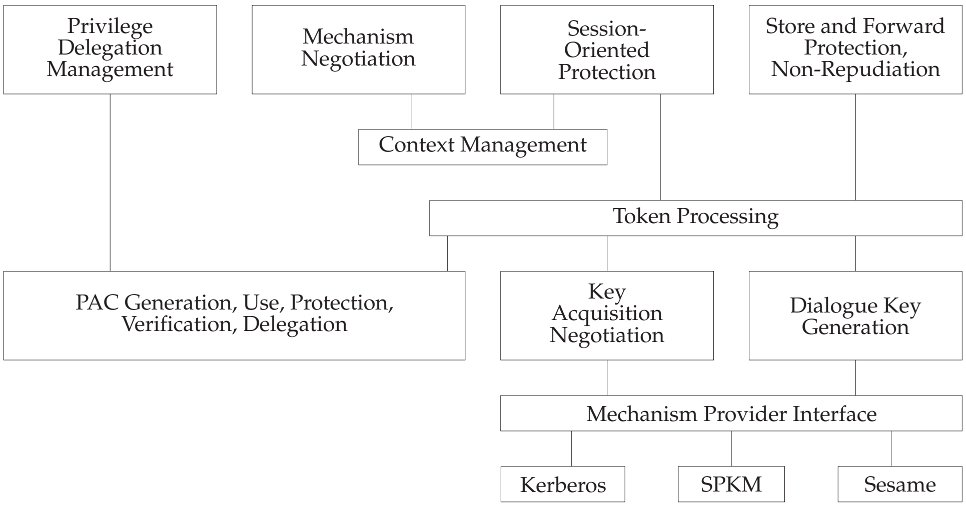

Figure 3-9 Protocol Security Service Structure

Session-Oriented Protocol Security Services

The preferred interface for session-oriented protocol security services is IETF RFC 2078: The GSS-API, Version 2. (The C bindings for this specification are defined in Generic Security Service API, Version 2: C Bindings.)

Store-and-Forward Protocol Security Services

The preferred interface for store-and-forward protocol security services is IETF RFC 2479: IDUP-GSS-API.

Non-Repudiation Services

The preferred interface for non-repudiation services is IETF RFC 2479: IDUP-GSS-API.

Recommendation: In addition to the interfaces described above, the PKI Task Group recommends that interfaces for protection mechanism negotiation and privilege and delegation management should be standardized. The preferred interfaces for these services are IETF RFC 2478: The Simple and Protected GSSI Negotiation Mechanism and IETF CAT XGSS-API, respectively.

Other interfaces which may support some or all of the protocol security services functionality include:

- Microsoft SSPI
- OMG CORBA Security
- TIPEM
- SHTTP

3.5.4 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 (for example, strength of supported confidentiality protection, delegation mode, and so on) 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 PKI Task Group is not aware of a draft profile in this area.

3.5.5 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 RFC 2478: The Simple and Protected GSSI Negotiation Mechanism.

3.6 Secure Protocol Components

There are many kinds of secure protocol, of which three important categories 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 3-10 illustrates the secure protocol components:

Figure 3-10 Secure Protocol Components

3.6.1 Function

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

3.6.2 Protocols

Examples of PKI-secured protocols include:

- *Connection-oriented peer-to-peer*: ONC GSS RPC, TLS, SHTTP, HTTPS, OMG SECIOP
- *Connectionless peer-to-peer*: IPsec
- *Connectionless multicast*: S/MIME, Open PGP, Secure multicast

Note: The IETF IPsec Working Group has produced work on secure protocols, notably IPsec (see **Referenced Documents** on page xiii) and IKE (see IETF RFC 2409: The Internet Key Exchange (IKE)).

3.6.3 Interfaces

Each secure protocol typically has its own interface. IETF RFC 2025: The Simple Public-Key GSS-API Mechanism (SPKM) gives an API for connection-oriented peer-to-peer services.

3.6.4 Profiles

It is not yet clear whether profiles will be established for which Web transaction security protocols (for example, SHTTP, HTTPS, HTTP-over-GSS-API, and so on) should be used in which contexts.

3.6.5 Negotiation

The PKI Task Group considers that negotiation of secure protocols is outside the scope of the PKI (or even Security Infrastructure) effort.

3.7 Security Policy Services Components

Figure 3-11 illustrates the security policy service components:

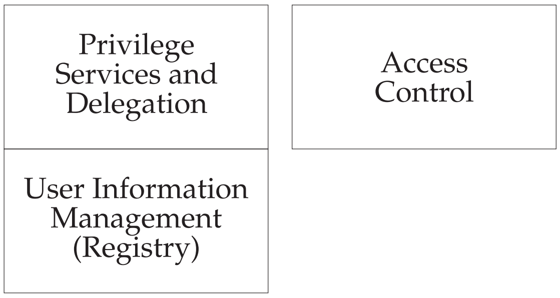

Figure 3-11 Security Policy Service Components

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.

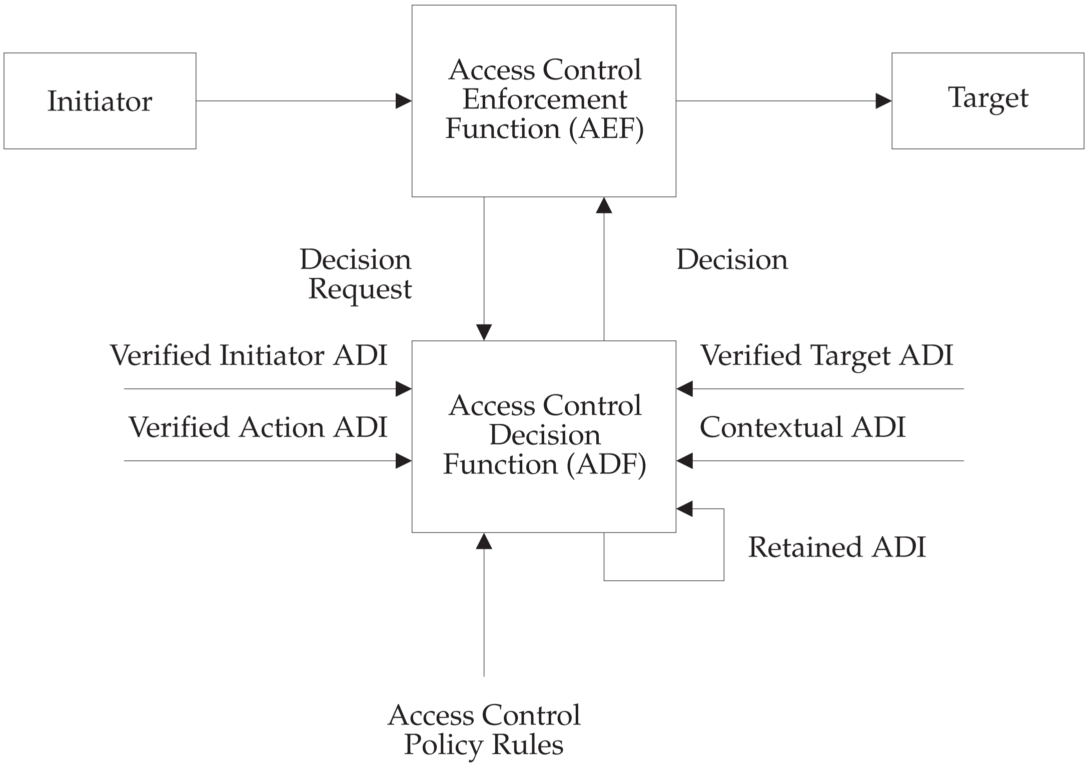

Figure 3-12 Access Control Decision Model

Figure 3-12 illustrates a model of access control enforcement within a system. See the Distributed Security Framework (XDSF) for a more detailed description.

An access request is mediated by the Access Control Enforcement Function (AEF). The AEF invokes the Access Decision Function (ADF) which makes a decision on the basis of the Access Decision Information (ADI) supplied to it by the AEF, or retrieved by the ADF from the processing context. The access decision is returned to the AEF which is responsible for enforcing the decision.

The ADI associated with a user principal is a subset of the privilege attributes assigned to the principal as part of user information management or associated with the principal through a process of delegation during operational use.

The privilege attribute service provides the functionality for the retrieval of ADI associated with a principal. Within a PKI Architecture, the ADI will be supplied in the form of privilege attribute security tokens.

There are two models of interaction with the privilege attribute service:

- *Push model*: in this model the privilege attributes are retrieved from the privilege attribute service by the initiator—for example, as part of initial user login—and then supplied (pushed) to the target service within the security certificate supplied by the initiator or a separate privilege attribute security token.

  The target service is responsible for verifying the authenticity of the supplied token. See Figure 3-13.

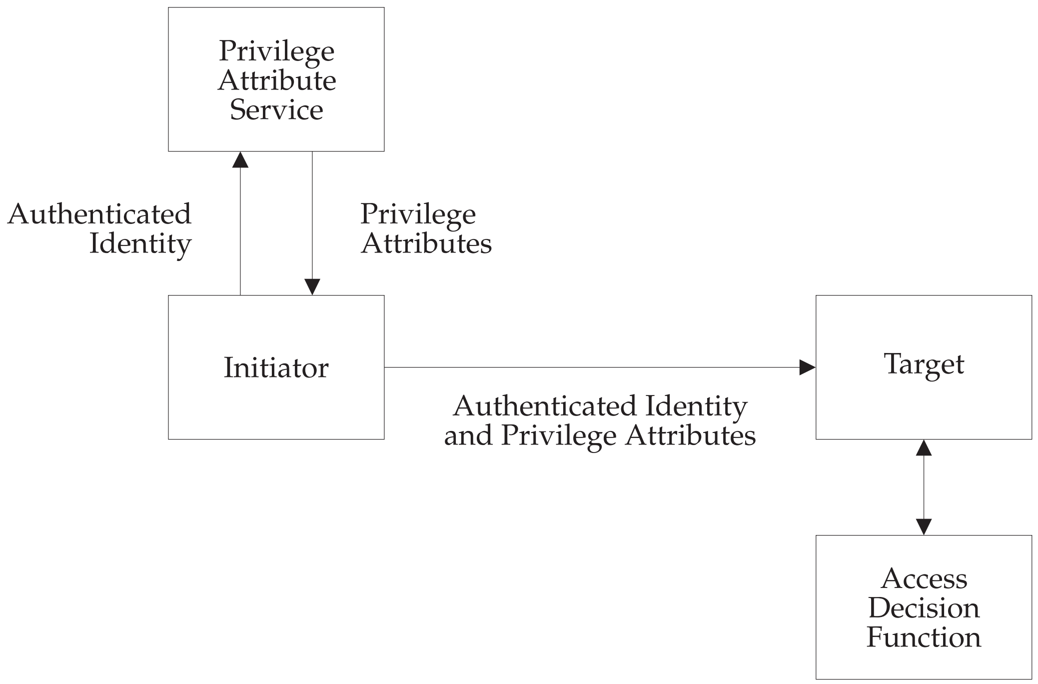

Figure 3-13 Push Model of Access Control

- *Pull model*: in this model the privilege attributes for the initiating principal are retrieved from the privilege attribute service by the target of the access request using the authenticated identity of the principal as supplied within an authentication certificate. See Figure 3-14.

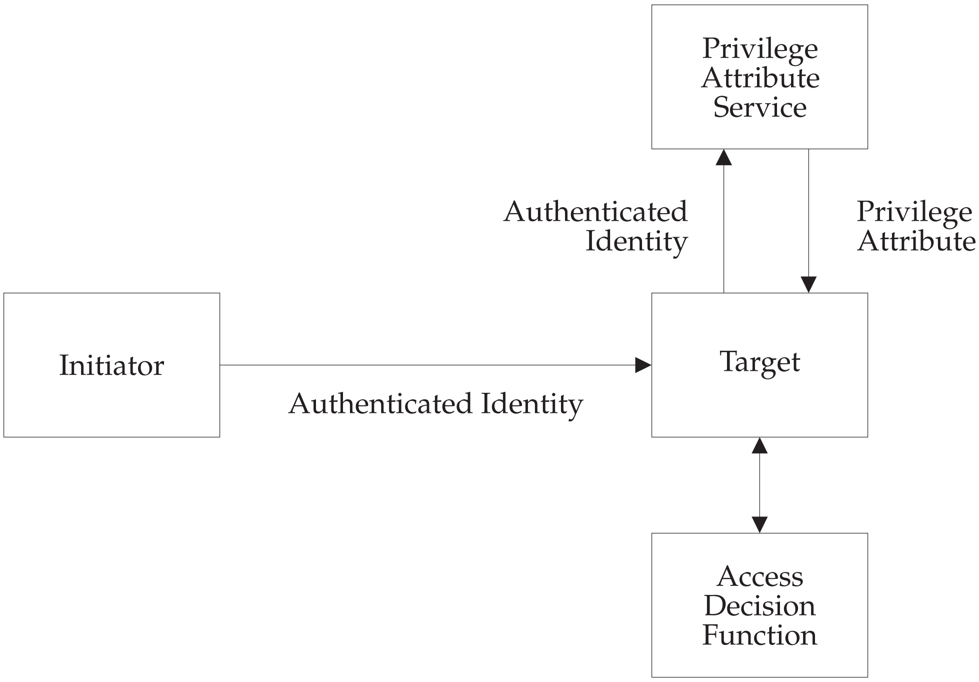

Figure 3-14 Pull Model of Access Control

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 PKI will define interfaces to privilege attribute services or access control services.

Work is currently underway within The Open Group on authorization services, and these may be the subject of future revisions to this document or associated architecture documents.

3.7.4 Profiles

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

3.8 Supporting Services Components

Figure 3-15 lists the supporting services components:

Figure 3-15 Supporting Services Components

3.8.1 Function

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

Security Auditing Services

The security auditing services support accountability within the PKI Architecture and may also support notarization services.

Time Service

The time service is fundamental to the synchronization of time within a distributed PKI Architecture and the basis of time stamps that may be incorporated into security certificates and also be used by a notarization service.

Directory Services

Directory services are necessary to support the location of PKI Architecture users and components and the retrieval of attributes applicable to them.

3.8.2 Protocols

Security Audit Service

Effective security auditing within a PKI Architecture requires the collection and analysis of security audit records from all components of the PKI Architecture. A standard protocol is needed to support a distributed security audit service between components of the PKI Architecture.

Recommendation: The Distributed Audit Service (XDAS) specification defines an API as well as a common audit record format that is applicable.

Time Service

A distributed time service needs a standard protocol to be defined to support the distribution and synchronization of time between the components of PKI.

Recommendation: It is recommended that IETF RFC 2030: Simple Network Time Protocol (SNTP) is adopted as the time service protocol for use within the PKI Architecture.

Directory Services

Protocols are needed to support the enquiry and retrieval of principal identities and attributes within the PKI Architecture. There are a number of candidate protocols:

- LDAPv2/LDAPv3
- ONC NIS, ONC NIS+
- XDS-simplified LDAP

Recommendation: It is recommended that IETF RFC 2251: Lightweight Directory Access Protocol, Version 3 (LDAPv3) is used as the basic directory service protocol within the PKI Architecture.

3.8.3 Interfaces

Security Auditing Service

An interface standard is required to enable components of the PKI Architecture to submit security audit records for security events they detect within their own operations. An interface standard is also required to support the development of applications for the collation and analysis of security audit records from all the components of the PKI Architecture.

Recommendation: It is recommended that the Distributed Audit Service (XDAS) specification is adopted as the security auditing service API.

Time Service

Components of the PKI Architecture will access time via the interface provided by the supporting operating system.

Directory Services

Recommendation: It is recommended that the C LDAP Application Program Interface is used as the interface to directory services.

3.8.4 Profiles

Profiles are currently not defined in the areas of audit and time services. Profiles for directory-based services are being developed within both The Open Group and IETF based upon LDAP. See *http://www.opengroup.org/ldap*.

Chapter 4

Hardware Security Devices in the PKI Architecture

It is expected that this section will be revised in a future version of the PKI Architecture to reflect current work within The Open Group on biometric and other authentication devices in conjunction with PKI and SSO architectures.

The PKI Architecture is intended to support at least two kinds of hardware security device: security tokens and cryptographic modules.

4.1 Security Tokens

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

Smartcards are assumed by the PKI Architecture to provide Virtual Smartcard services. They will also frequently also provide at least the key activation and signing components of cryptographic services; they may also provide other cryptographic services.

Memory cards provide only storage; Virtual Smartcard services involving state maintenance (for example, 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.

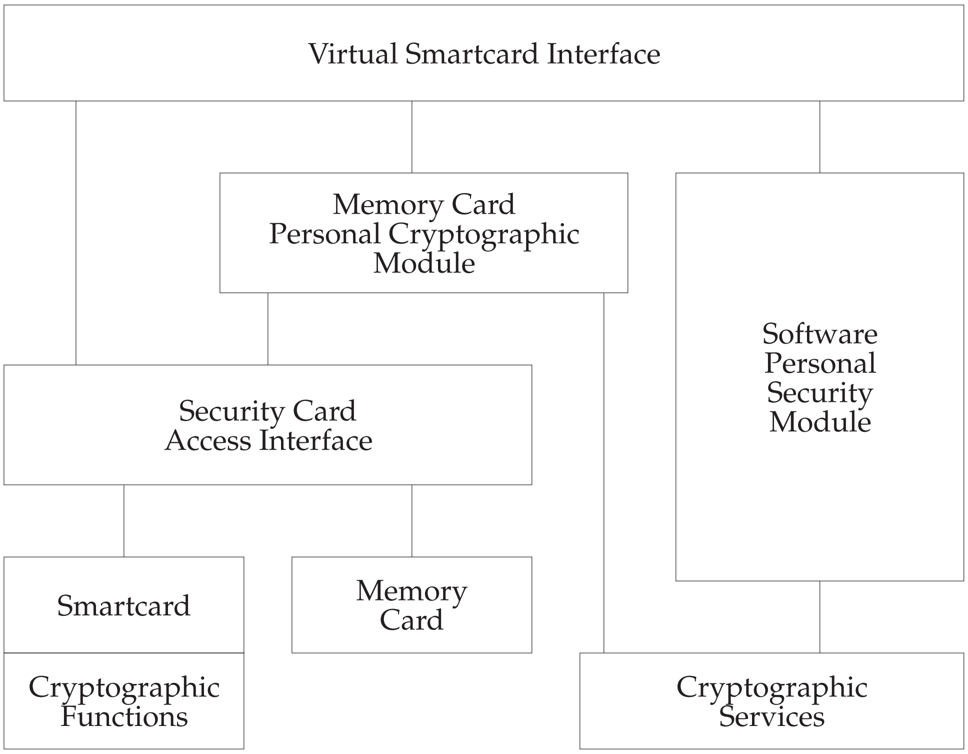

Figure 4-1 Hardware Security Devices

Time-synchronized and challenge-response tokens provide only authentication functionality, and will typically be integrated into the PKI Architecture through modifications to the system security-enabling services (particularly the logon and obtain credentials components of those services).

4.2 Cryptographic Modules

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

Cryptographic modules are assumed by the PKI Architecture to provide the full range of cryptographic services (and they may provide direct access to some cryptographic primitives for the convenience of designers of new cryptographic services).

Appendix A

Requirements for Virtual Smartcard Services

A feature of PKI services is a need to store long-term personal security information (including private keys, certificates, and other information) in protected storage, to activate personal keys for use via an authentication procedure, and to use those keys for encryption, decryption, and signature activities.

There are two models for the processing and management of this information:

- One model (exemplified by PGP and Lotus Notes) manages private keys and personal data 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).
- The second model (exemplified by ONC RPC and Novell NetWare) manages private keys and personal data at a central server and distributes them to client principals using a secure protocol.

The first model may be supported by Smartcard technology in which the personal data is processed and managed within a separate hardware device (the Smartcard). However, the use of such Smartcards incurs the cost of additional hardware and software. They are therefore used in circumstances when the additional security they provide justifies the extra cost. An alternative approach is to use a software module in those circumstances when the additional security is not cost-justified.

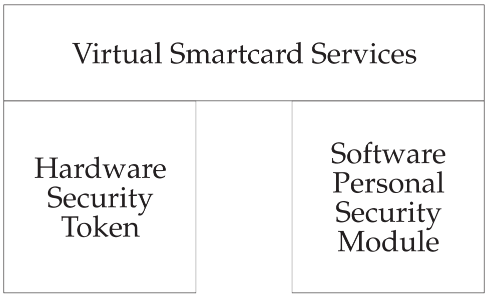

Figure A-1 Virtual Smartcard Service Structure

From an application perspective it is not relevant whether a hardware or software module is used, provided the requisite data and services may be accessed. Hence the concept of a Virtual Smartcard service, which may be layered over either hardware or software implementations, may be envisaged within a system architecture. Figure A-1 illustrates the structure of this component.

The concept of a software module may be further extended to encompass services provided under the second model, based on a centralized repository. 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 A-2, (the dotted arrow in the figure represents the protocol):

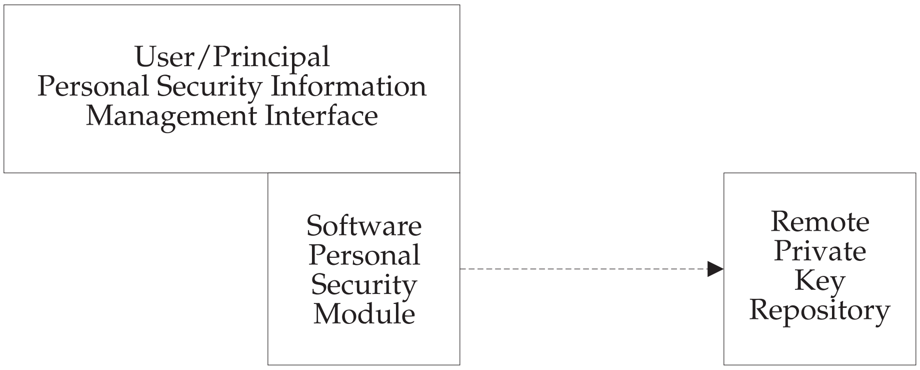

Figure A-2 Virtual Smartcard Service Protocol

The concept of a Virtual Smartcard service may be extended to encompass the processing and management of all personal security information, whichever model of implementation is used.

The Virtual Smartcard service will contribute to generic solutions for end users who are facing real situations that can be summarized as follows:

- **Facilitate the handling of information related to multiple identities in a heterogeneous environment:**
  - As members of an enterprise (corporate users), end users may have multiple identities depending on systems and applications they need to access. Whenever Single Sign-on is achieved in their enterprise IT, the multiplicity of identities continues to exist despite being hidden behind a single identity. An example of end-user identities is login names.
  - End users must be authenticated by heterogeneous applications in security infrastructures which combine legacy technologies (based on passwords, for instance) with public-key-based technologies.
  - As consumers of distributed services over the Internet (Internet users), end users do not use the same identity when they access, for example, a home banking service or an electronic commerce service. The Virtual Smartcard service is designed to support the handling of multiple identities by end users in the electronic commerce market place.
  - Permit users to store and access long-term personal security information certified by more than a single certification authority (handle multiple identities, and so on).
- **To provide secure means of handling and managing personal data:**
  - End users own data that can be qualified as personal information because it is data given to them, acquired by them, or generated by them. Certificates, passwords, private keys, or even identities are examples of this kind of data.
  - Permits users and other principals to store long-term personal security information (including private keys, certificates, and other information) in protected storage, to activate personal keys for use via an authentication procedure, and to use those keys for encryption, decryption, and signature activities.
- **To support end-user mobility:**

  End users are more and more mobile either within intranets or across them, or over the Internet. Therefore, they want to be able to transport their personal information securely.

- **Avoid the need for and cost of uniform use of hardware devices:**

  Support for the deployment of Smartcard techniques without the need to have Smartcard hardware implemented on every workstation.

- **To be hardware-independent:**

  — Support different hardware or software devices such as Smartcards, floppies, PCMCIA cards, or software modules

  — Support mobile devices such as Smartcards

  — Support multiple Smartcard technologies (which can be multi-application)

  — Support multi-PIN systems

  — Provide flexibility to accommodate the increasing functionality expected to be implemented within Smartcards as the technology evolves. The Virtual Smartcard service can facilitate this by enabling a varying mixture of software and hardware service implementations below the interface.

The following problem scenarios illustrate issues regarding the management of the information contained on Smartcards:

- Temporary access to services:

  What happens when someone leaves their Smartcard at home (enterprise environment)?

- Management of long-lived data:

  — Smartcards may have a limited lifetime for reasons of security, or just physical wear and tear

  — As functionality evolves, there is still a need to be able to preserve data

  — Hence there is a general requirement for export/import functionality which enables transfer of a key and other data from one Smartcard to another

A.1 Overview of Virtual Smartcard Services

A Virtual Smartcard service comprises three aspects:

1. A configuration capability that allows transparent handling of diverse hardware or software security devices. These devices will be used as personal data repositories or providers of security mechanisms.
2. An abstract data model through which personal information is accessed. This data model describes personal data in a two-level hierarchy. The first level consists of personal data domains (application domains or security domains); the second level consists of end users' identities.
3. A set of functions designed to:
   — Customize and initialize the data model
   — Configure the security devices where the personal data resides
   — Manipulate personal data through the data model

Figure A-3 illustrates the type of data and a data structure that might be maintained by a Virtual Smartcard service.

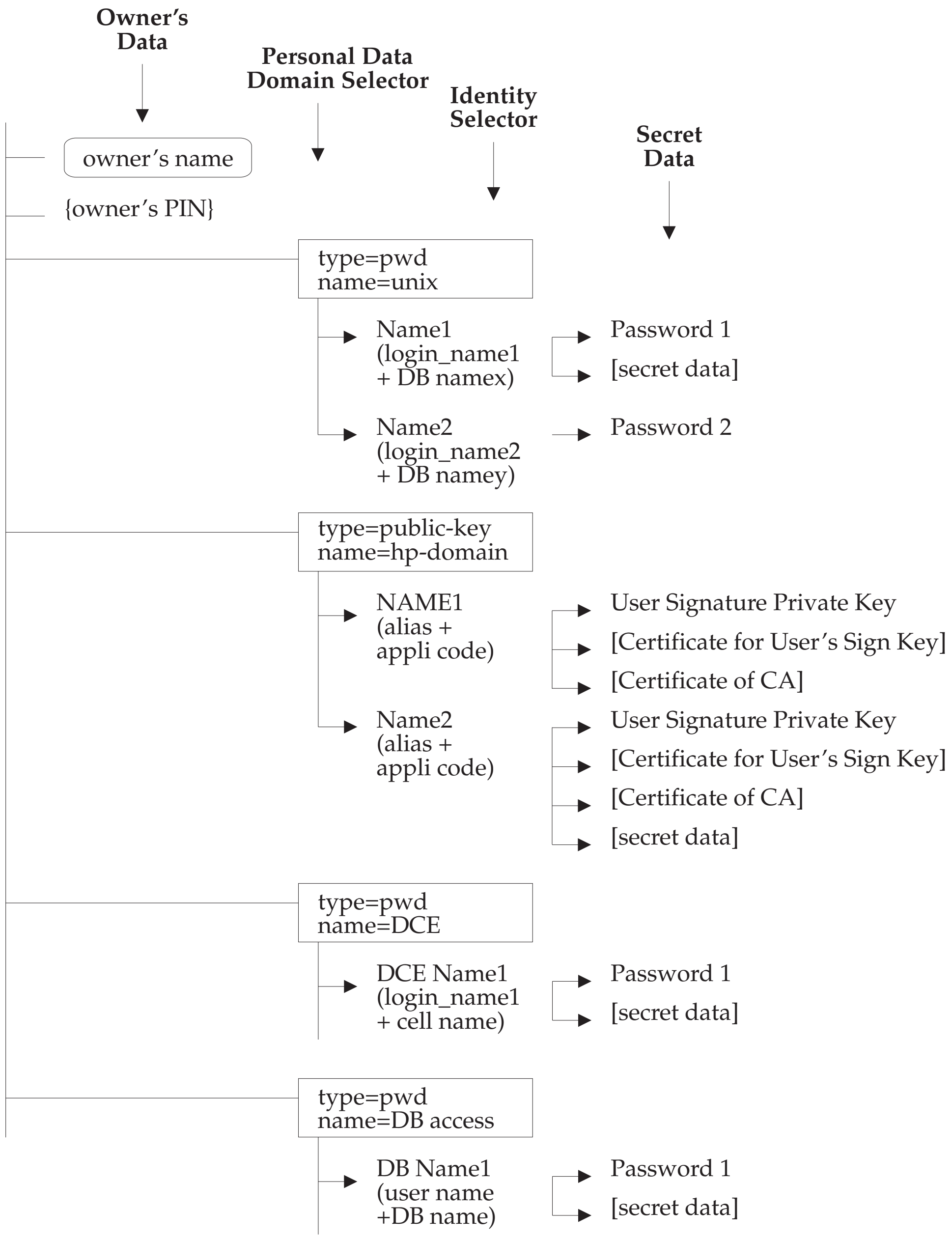

Figure A-3 Example Data

Glossary

access control
The prevention of unauthorized use of a resource including the prevention of use of a resource in an unauthorized manner (see ISO/IEC 7498-2).

access control policy
The set of rules that define the conditions under which an access may take place (see ISO/IEC 10181-3).

accountability
The property that ensures that the actions of an entity may be traced to that entity (see ISO/IEC 7498-2).

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.

association-security-state
The collection of information that is relevant to the control of communications security for a particular application-association (see ISO/IEC 10745).

audit
See **security audit** (see ISO/IEC 7498-2).

audit trail
See **security audit trail** (see ISO/IEC 7498-2).

authenticated identity
An identity of a principal that has been assured through authentication (see ISO/IEC 10181-2).

authentication
Verify claimed identity; see **data origin authentication** and **peer-entity authentication** (see ISO/IEC 7498-2).

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 ISO/IEC 10181-2).

authentication exchange
A sequence of one or more transfers of exchange authentication information (AI) for the purposes of performing an authentication (see ISO/IEC 10181-2).

authentication information (AI)
Information used to establish the validity of a claimed identity (see ISO/IEC 7498-2).

authorization
The granting of rights, which includes the granting of access based on access rights (see ISO/IEC 7498-2).

availability
The property of being accessible and usable upon demand by an authorized entity (see ISO/IEC 7498-2).

certification authority (CA)
A certification authority issues certificates and vouches for the identities of those individuals or entities to whom it issues certificates, and their association with a given key.

ciphertext
Data produced through the use of encipherment. The semantic content of the resulting data is not available (see ISO/IEC 7498-2).

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

clear text
Intelligible data, the semantic content of which is available (see ISO/IEC 7498-2).

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 unauthorized individuals, entities, or processes (see ISO/IEC 7498-2).

contextual information
Information derived from the context in which an access is made (for example, time of day) (see ISO/IEC 10181-3).

credentials
Data that is transferred to establish the claimed identity of an entity (see ISO/IEC 7498-2).

cryptanalysis
The analysis of a cryptographic system and its inputs and outputs to derive confidential variables and/or sensitive data including clear text (see ISO/IEC 7498-2).

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 ISO/IEC 7498-2).

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 unauthorized use (see ISO/IEC 7498-2).

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

data integrity
The property that data has not been altered or destroyed in an unauthorized manner (see ISO/IEC 7498-2).

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 ISO/IEC 7498-2).

decryption
See **decipherment** (see ISO/IEC 7498-2).

denial of service
The unauthorized prevention of authorized access to resources or the delaying of time-critical operations (see ISO/IEC 7498-2).

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 ISO/IEC 7498-2).

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 ISO/IEC 10181-2).

encipherment
The cryptographic transformation of data (see **cryptography**) to produce ciphertext (see ISO/IEC 7498-2).

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 ISO/IEC 7498-2).

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 ISO/IEC 7498-2).

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).

integrity
See **data integrity** (see ISO/IEC 7498-2).

key
A sequence of symbols that controls the operations of encipherment and decipherment (see ISO/IEC 7498-2).

key management
The generation, storage, distribution, deletion, archiving, and application of keys in accordance with a security policy (see ISO/IEC 7498-2).

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

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 ISO/IEC 10181-2).

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 ISO/IEC 10181-2).

password
Confidential authentication information, usually composed of a string of characters (see ISO/IEC 7498-2).

peer-entity authentication
The corroboration that a peer entity in an association is the one claimed (see ISO/IEC 7498-2).

principal
An entity whose identity can be authenticated (see ISO/IEC 10181-2).

private key
A key used in an asymmetric algorithm. Possession of this key is restricted, usually to only one entity (see ISO/IEC 10181-1).

public key
The key, used in an asymmetric algorithm, that is publicly available (see ISO/IEC 10181-1).

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 ISO/IEC 7498-2).

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 ISO/IEC 10181-1).

secret key
In a symmetric cryptographic algorithm the key shared between two entities (see ISO/IEC 10181-1).

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.

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

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 ISO/IEC 7498-2).

security audit trail
Data collected and potentially used to facilitate a security audit (see ISO/IEC 7498-2).

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 ISO/IEC 10181-1).

Note: All certificates are deemed to be security certificates (see the relevant definitions in ISO/IEC 7498-2) adopted in order to avoid terminology conflicts with the directory authentication standard.

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 ISO/IEC 7498-2).

Note: The marking may be explicit or implicit.

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.

signature
See **digital signature** (see ISO/IEC 7498-2).

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 ISO/IEC 10181-1).

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 ISO/IEC 7498-2).

trusted third party

A security authority or its agent, trusted by other entities with respect to security-related operations (see ISO/IEC 10181-1).

Index