REQUIREMENTS FOR KEY RECOVERY PRODUCTS

(ADVISORY COMMITTEE WORKING DRAFT)

Available at http://csrc.nist.gov/keyrecovery/

This is a working draft of the Technical Advisory Committee to Develop a Federal Information Processing Standard for the Federal Key Management Infrastructure (TAC). As such, this document has not been drafted, approved or adopted by the Federal Government.

Foreword

The Federal Information Processing Standards Publication Series of the National Institute of Standards and Technology (NIST) is the official publication relating to standards and guidelines adopted and promulgated under the provisions of Section 5131 of the Information Technology Management Reform Act of 1996, and the Computer Security Act of 1987, Public Law 104-106. Under these mandates, the Secretary of Commerce promulgates standards and guidance pertaining to the efficiency, security and privacy of Federal computer systems. The National Institute of Standards and Technology, through its Information Technology Laboratory, has the mission of developing standards, guidelines and associated methods and techniques for computer systems, and providing technical assistance to industry and government in the implementation of standards.

Comments concerning Federal Information Processing Standards Publications are welcomed and should be addressed to the Director, Information Technology Laboratory, National Institute of Standards and Technology, Gaithersburg, MD 20899.

Shukri Wakid, Director
Information Technology Laboratory

Abstract

This standard specifies requirements to be met by government Key Recovery Systems. Such systems provide for the decryption of stored or communicated data when access to the data is properly authorized.

ALTERNATIVE TO THE ABOVE: This standard specifies requirements to be met by key recovery products used by Federal government agencies. These products provide for the recovery of keys which will be used for the decryption of stored or communicated data when access to the data is properly authorized.

Key words: ADP security, computer security, Key Recovery, Federal Information Processing Standard.

Federal Information
Processing Standards Publication XXX

(Date)

Announcing the

REQUIREMENTS FOR KEY RECOVERY PRODUCTS

Federal Information Processing Standards Publications (FIPS PUBS) are issued by the National Institute of Standards and Technology (NIST) after approval by the Secretary of Commerce pursuant to Section 5131 of the Information Technology Management Reform Act of 1996, and the Computer Security Act of 1987, Public Law 104-106.

1. Name of Standard. Requirements for Key Recovery Products.

2. Category of Standard. Computer Security, Cryptography.

3. Explanation. This Standard specifies requirements for key recovery products. These products provide for the recovery of keys to be used for the decryption of stored or communicated ciphertext when the decryption keys are not otherwise available. Key recovery is motivated by three primary scenarios:

1. recovery of stored data on behalf of an organization (or individual) e.g., in response to the accidental loss of keys;
2. recovery of stored or communicated data on behalf of an organization (e.g., for the purposes of monitoring or auditing activities); and
3. recovery of communicated or stored data by authorized authorities.

The first scenario supports the ability to regain access to data that would otherwise be lost. The second scenario encompasses internal investigation authorized by an organization. The final scenario encompasses data acquired under the authorization of court orders for wiretaps, search and seizure orders, civil suit subpoenas, etc

A Key Recovery System (KRS) manages cryptographic keys in support of data recovery when normal key access mechanisms fail. These systems must be carefully designed so that plaintext may be recovered in a timely manner, and so that only authorized recoveries are permitted. Therefore, security is a critical factor in any KRS design.

The purpose of this standard is to specify requirements for key recovery products, and to enable the validation of products claiming conformance. The standard encompasses the functional, security, assurance and interoperability of key recovery products.

4. Approving Authority. Secretary of Commerce.

5. Maintenance Agency. U.S. Department of Commerce, National Institute of Standards and Technology (NIST), Information Technology Laboratory.

6. Cross Index.

a. FIPS PUB 46-2, Data Encryption Standard.
b. FIPS PUB 81, DES Modes of Operation.
c. FIPS PUB 140-1, Security Requirements for Cryptographic Modules, January 1994.
d. DOD 5200.28-STD, Department of Defense Trusted Computer System Evaluation Criteria (TCSEC) ("The Orange Book"), National Computer Security Center, December 1985.
e. SC 27 N1953, Evaluation Criteria for IT Security, Part 3 – Security Assurance Requirements
f. ISO 7498-2, Information Processing Systems - Open System Interconnection -Basic Reference Model - Part 2: Security Architecture; February 1989.

Other NIST publications may be applicable to the implementation and use of this standard. A list (NIST Publications List 91) of currently available computer security publications, including ordering information, can be obtained from NIST.

7. Applicability. To be supplied by the Federal Government.

8. Applications. This standard is appropriate for use in a variety of applications, including (but not limited to):

1. When computer files are encrypted for secure storage or transmission,~~;~~
2. When electronic mail is encrypted before transmission among communicating entities,~~;~~ ~~and~~
3. When electronic voice,~~-~~ fax , or video communications are encrypted for privacy, and
4. When link or network layer encryption is employed to provide bulk protection.

9. Specifications. Federal Information Processing Standard (FIPS xyz) Requirements for Key Recovery Products (affixed).

10. Implementations. Implementations of this standard may be in software, firmware, hardware, or any combination thereof. All cryptographic modules employed in such implementations shall comply with FIPS 140-1. FIPS approved encryption algorithms (e.g., DES) shall be used in

Federal applications of systems conforming to this standard. The use of new encryption algorithms which are FIPS approved after the date of the standard is also permitted.

Information about the validation of implementations conforming to this standard may be obtained from the National Institute of Standards and Technology, Information Technology Laboratory, Attn: Key Recovery Validation, Gaithersburg, MD 20899.

11. Export Control. To be supplied by the Federal Government.

12. Patents. Implementations of this standard may be covered by U.S. and foreign patents.

13. Implementation Schedule. To be supplied by the Federal Government.

14. Glossary. The following terms are used as defined below in this standard: [NOTE: THE GLOSSARY WAS NOT REVIEWED BY THE TAC]

Abstract Machine
The underlying hardware or firmware abstraction to which the software is written.

Accountability
The property that ensures that the actions of an entity ~~may~~ can be traced uniquely to the entity.

Assurance
~~(1) Confidence that an entity meets its security objectives. (2)~~ The degree of confidence that a product correctly implements the security ~~policy~~functions. In the context of this FIPS, three levels of assurance are specified, representing increasing degrees of confidence.

Auditable Events
~~Events~~ Security relevant machine transactions within a key recovery product which may appear in an audit log (see Section 4).

Authentication Data
Information used to ~~authenticate~~ verify the claimed identity of an entity, e.g., a password, PIN, biometric, or response to a challenge.

~~Authentication Information~~
~~See "Authentication Data".~~

Authentication Mechanism
A technique used to ~~authenticate~~ verify the claimed identity of an entity, e.g., user ID and password, token, biometric, or challenge-response.

?Authentic Public Key Source
~~Used to~~ An entity that provides a certificate infrastructure to support the use of public key cryptography within ~~the~~ a Key Recovery System.

?Authorized key recovery

Key recovery either with the permission of the owner of the data or as otherwise permitted by law.

?Authorized Request

A request based on a legal and lawful right for access by a data owner or other authorized entity.

Authorized User

A user who is authorized to access a system to perform one or more ~~actions~~operations.

Common Criteria (CC)

An international standard for security in information security products. (See Cross Index.)

Common Criteria Evaluation Assurance Level (EAL)

One of ~~A~~a predefined set of assurance requirements ~~products that represents a point on~~ from the Common Criteria ~~assurance scale~~.

Common Criteria Protection Profile

An implementation-independent set of security requirements for a category of products ~~which~~ that meet specific consumer needs.

Confidentiality

~~(1) Assurance that the information is not disclosed to unauthorized entities or processes. (2) The property that sensitive information is not disclosed to unauthorized individuals, entities or processes. (3)~~ The property that information is not made available or disclosed to an unauthorized user, process, or object.

Configurable ~~Capability~~

A ~~capability~~ feature of a product that may or may not be enabled~~is available but need not be selected for use~~.

Configuration Item

An ~~I~~item~~s~~ (e.g., documents, software, hardware) ~~which are~~ under configuration control.

Configuration Management (CM)

The management of security features and assurances through the control of changes made to a system's hardware, software, firmware, documentation set, test, test fixtures, and test documentation throughout the development and operational life of the system.

Cryptographic End System (Recoverable)

~~A system containing encryption and decryption mechanisms. Incorporates a KRI Generation, KRI Delivery or KRI Validation Function.~~ See section 2.6.

Cryptographic

~~The~~ A set of hardware, firmware, software, or some combination thereof

Module (cryptomodule)
that implements cryptographic logic, cryptographic processes, or both.

Data
Voice, facsimile, computer files, electronic mail, and other stored or communicated information.

Data Encryption Key (DEK)
A ~~symmetric~~ cryptographic key used to encrypt data. In a symmetric cryptosystem, the same (or an easily derived) key also is used to decrypt data.

Data origin authentication
The ability to authenticate the identity of the source of information. See ISO 7498-2.

Data Recovery System
The system/subsystem used to recover encrypted data using a recovered target key obtained by ~~the~~ a Key Recovery Requestor ~~System~~ function.

Decryption
~~(1) Transformation of ciphertext form of data to plaintext form. (2) The~~ A process ~~of~~ for ~~changing~~ transforming ciphertext into plaintext, using a cryptographic algorithm and a key.

Encryption
~~(1) Transformation of plaintext form of data to ciphertext form. (2) A process of transforming plaintext into ciphertext for the purpose of security or privacy. (3) Transforming text into code in order to conceal its meaning. The process of transforming data to an unintelligible form in such a way that the original data either cannot be obtained (one-way encryption), or cannot be obtained without using the inverse decryption process. (3) Conversion~~ A process for transforming~~of~~ plaintext to ciphertext through the use of a cryptographic algorithm and a key.

Evidence of Origin
~~(1)~~ A proof of the origin of information that cannot be ~~refuted~~ (successfully) repudiated by the originator, e.g., ~~by~~ a message digitally signed by the originator.~~using a digital signature~~. ~~(2) Non-repudiation.~~

Evidence of Receipt
A proof of the receipt of information ~~so~~ that cannot be (successfully) repudiated by the recipient ~~cannot deny having received the information~~, e.g., a digitally signed receipt issued ~~using a digital signature~~ by the recipient of~~n~~ the ~~received~~ message.

FIPS 140-1 ~~Level 1 Security Requirements~~
This FIPS ~~S~~specifies~~y~~ ~~basic~~ security functionality and assurance requirements for ~~a~~ cryptomodules. ~~No physical security mechanisms are required in the module beyond the requirement for production-grade equipment. Software cryptographic functions may be performed in a general purpose personal computer.~~ See Cross Index.

~~FIPS 140-1 Level 2 Security Requirements~~ ~~Improves upon the physical security of a Level 1 cryptomodule by (1) requiring tamper evident coatings or seals, or for pick-resistant locks, (2) requiring role-based authentication and (3) allowing software cryptography in multi-user timeshared systems when used in conjunction with a C2[1] or equivalent operating system.~~

~~FIPS 140-1 Level 3 Security Requirements~~ ~~Improves upon the Level 1 and 2 requirements for cryptomodules by (1) requiring tamper detection mechanisms, (2) requiring identity-based authentication, (3) specifying stronger requirements for entering and outputting critical security parameters, and (4) allowing software cryptography in multi-user timeshared systems when a B1 or equivalent trusted operating system is employed along with a trusted path for the entry and output of critical security parameters.~~

FIPS Compliant Meeting all requirements of a specified level of ~~this~~ a FIPS~~standard~~.

Flaw Remediation The correction of discovered security flaws in a product or system.

Functional Requirements A high level description of the requirements for a product or system.

~~Functional Specification~~ ~~High level description of the user-visible interface and behavior of a system.~~

Implementation Representation A description of the implementation (e.g., source code when the implementation is software or firmware; or drawings and schematics, if the system is hardware).

Independent Testing Testing performed by persons other than the developers.

Informal Security Policy Model An accurate and concise statement of system security policy expressed ~~informally (i.e.,~~ in natural language,~~;~~ e.g., English~~)~~.

~~Informal~~ ~~(1) Expressed in natural language. (2) Written as prose in natural language.~~

~~Informal style/presentation~~ ~~Written in normal language, e.g., English~~.

[1] The C2, B1 and B2 ratings are in accordance with the TCSEC (see the cross index in the Announcement section).

Integrity	The property that ~~sensitive~~ data has not been modified ~~or deleted~~ in an unauthorized and undetected manner.
~~Interactive Communication~~	~~Two-way communication between end users.~~
Interoperability	The ability of products or systems to communicate with one another.
Key Escrow	~~(1) The processes of managing (e.g., generating, storing, transferring, auditing) the cryptographic keys or key products by one or more entities. (2) A key recovery technique that employs one or more Key Recovery Agents who hold (i.e., cache) keys or key products for their subscribers. (3)~~ A method of key recovery in which~~where~~ the secret or private keys, key parts, or key related information to be recovered is stored by one or more Key Recovery Agents. ~~Other Key Recovery Information may be available elsewhere.~~
~~Key Recovery~~	~~Access to information sufficient to recover encrypted data.~~
Key Recovery Agent (KRA) Function	A key recovery system function that performs a recovery service in response to an authorized request ~~by a requestor system on behalf of a requestor~~.
~~Key Recovery Agent Function~~	~~Performs a key recovery service in response to an authorized request.~~
~~Key Recovery Capable System~~	~~A cryptographic end system with either a KRI Generation Function or a Key Recovery Validation Function or both.~~
Key Recovery Product	A product that performs one or more key recovery system functions.
~~Key Recovery Field~~	~~A field, output by the key recovery mechanism of a product, that identifies key recovery agents and enables key recovery agents to identify the key(s) required to decrypt corresponding ciphertext output by the product.~~
Key Recovery Information (KRI)	All or part of the ~~required~~ information that is ~~used in~~required for the recovery of a key. The KRI does not include a plaintext key.
~~Key Recovery~~	~~Key recovery information which is specific to a single key recovery~~

~~Information Field (KRIF)~~

~~scheme.~~

Key Recovery Block (KRB)

A ~~stream of bytes~~data structure that serves as a container for a single key recovery scheme-specific KRI~~F~~ and associates the KRI~~F~~ with a set of standard fields in a predefined format.

Key Recovery Policy

A policy ~~which~~ that specifies the conditions under which key recovery information must be created and conditions under which and to whom the key recovery information may be released; it may also indicate the allowable Key Recovery Agent(s) and how or where key recovery information must be maintained.

Key Recovery Requestor (KRR) ~~Function~~Function

~~The~~ A function in a key recovery system ~~system/subsystem~~ used ~~by the requestor~~ to request keys.

~~Key Recovery Service~~

~~An authorized key recovery as performed by a Key Recovery Agent.~~

Key Recovery System (KRS)

~~Consists~~ This consists of the KRI Generation Function, the KRI Management Function, and the Key Recovery Function. It includes software, hardware, procedures, and infrastructure.

~~KRA~~

~~Key Recovery Agent~~

~~KRB~~

~~Key Recovery Block~~

KRI Delivery Function

A key recovery system function ~~A~~assembles and formats ~~the~~ key recovery information (KRI) and makes ~~the~~ it ~~KRI~~ available for recovery and validation.

~~KRI~~ Key Encapsulation

A method of key recovery in which keys, key parts, or key related information is encrypted specifically for the KRA Function and associated ~~maintained~~ with the encrypted data~~outside a Key Recovery Agent~~.

KRI Generation Function

A key recovery system function that ~~G~~generates all or part of the key recovery information (KRI)~~ needed to recover the target key and provides the KRI to the KRI Delivery Function~~.

~~KRI Providers~~

~~Those entities provide Key Recovery Information (KRI) using a KRI Generation Function.~~

KRI Validation Function

A key recovery system function that~~Checks,~~ authenticates~~, validates~~ or verifies ~~the available~~ key recovery information.

~~KRR~~

~~Key Recovery Requestor System.~~

~~KRS~~

~~Key Recovery System~~

~~Layered Product~~

~~A product in which security functions are layered. For example, a secure application which is implemented on top of a secure operating system is a layered product.~~

Least Abstract Representation

~~(1)~~ The most concrete representation of an implementation (e.g., source code). ~~(2) The representation that is closest to the implementation, e.g., source code.~~

~~Licensing Agent~~

~~Authorizes Key Recovery Agents after an evaluation against the FIPS.~~

~~Masquerading~~

~~An attempt to gain access to a system by posing as an authorized user.~~

~~Message Security Protocol (MSP)~~

~~A data format that cryptographically binds data sensitivity and provides public key cryptography based security services for the data, including confidentiality, integrity, etc.~~

~~MIME~~

~~Multipurpose Internet Mail Extension as defined in RFC 2045.~~

~~Monolithic Product~~

~~A product in which security functions are not layered. See "Layered Product".~~

~~Non-Key Recovery Product~~Cryptographic End System (non-recoverable)

An encryption product ~~whose encryption~~ the output of which is not recoverable~~ through key recovery~~.

Presentation of Evidence

Providing ~~the~~ information necessary to carrying out the assurance activity.

Private Key

~~(1) In an asymmetric (public) key cryptosystem, that key of an entity's key pair which is known only by that entity. (2)~~ A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and not made public.

Public Key

~~(1) In an asymmetric key system, that key of an entity's key pair which is~~

~~publicly known. (2)~~ A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public.

Recovery Registration Information (RRI?)~~Registration (at a KRA)~~

~~Deposit of target key information and other relevant information which will allow key recovery using the KRA.~~ Information provided to a KRA in support of (later) key recovery.

~~Registration Agent~~

~~Archives vendor-specific information in order to find, acquire and parse recovery information.~~

Representation Correspondence

An accurate and complete mapping from a higher level representation to a lower level representation (e.g., from functional requirements to a functional specification, from a functional specification to a high level design, from a high level design to a low level design, from a low level design to source code, etc.).

~~Requestor~~

~~An entity that is authorized to request a key recovery.~~

~~Requestor Subsystem~~

~~Interacts with one or more Key Recovery Agents using Key Recovery Information to recover a data encrypting key.~~

Secret Key

A cryptographic key used with a secret key [symmetric] cryptographic algorithm, ~~uniquely associated~~ known by~~with~~ one or more entities, and ~~which shall~~ not be made public.

Security Domain

~~(1) A set of objects , a security policy , a security authority and a set of relevant activities inwhich the set of elements are subject to the security policy , administered by the security authority , for the specified activities. (2)~~ A set of security-related services, mechanisms, and policies.

Security Policy

(~~1) A precise specification of the security rules under which a cryptographic module may operate, including the security rules derived from the requirements of this standard and the additional security rules imposed by the manufacturer. (2)~~ A set of rules and procedures regulating the use of information including its processing, storage, distribution, and presentation.

~~Security Policy Model~~

~~A formal representation of the security policy enforced by the product.~~

~~Security Target~~
~~A set of security requirements ad specifications to be used as the basis for evaluation of an identified product.~~

~~Session-based Protocols~~
~~Interactive communications.~~

~~Session Key~~
~~A key that is used to encrypt and/or decrypt data for a single communications session.~~

~~Session Key Recovery~~
~~Recovery of the Data Encryption Key.~~

~~S/MIME~~
~~Secure MIME as defined by RFC XXX.~~

Standard Communication Protocol
Any communication protocol adopted by a generally recognized standards organization. [pick up page 13 text]

~~Store-and-Forward Communications~~
~~One way communications (i.e., from a sender to a receiver) without the involvement of the receiver. The receiver may acquire the communication at a time which is significantly later than the time the communication is sent.~~

~~System~~
~~Includes software, hardware, procedures.~~

Target key
The cryptographic key recovered by a Key Recovery System.

Target key information
~~(1)~~ Information provided ~~held~~ by a KRA in response to an authorized key recovery request. ~~which is used to reconstruct a target key, e.g., the target key may be reconstructed by performing a mathematical calculation using one or more "pieces" of target key information.~~

~~Testing laboratory~~
~~A laboratory which has been accredited by NIST to test systems, subsystems, key recovery agents, or products for conformance to this standard.~~

~~Transaction-based Protocols~~
~~Store-and-forward communications.~~

Trusted Path
A mechanism by which a person or process can communicate directly with a ~~cryptographic module~~ key recovery system function and which can

~~only~~ be activated only by the person~~,~~ process, or the function.~~or module, and cannot be imitated by untrusted software within the module.~~

~~Trusted Third Party~~ ~~An entity which is trusted by the parties performing the encryption or decryption processes, but are not identical with those parties.~~

Trusted Time Stamp A date and time that is reliable, accurate, and is affixed in such a way as to preclude undetected modification. ~~that it can not be modified by parties other than the time stamping source without detection.~~

~~Unwrap~~ ~~Decryption of an encrypted key by another key.~~

~~Vulnerability Analysis~~ ~~The determination of the vulnerabilities of a product or system.~~

~~Wrap~~ ~~Encryption of a cryptographic key by another key.~~

15. Qualifications. The security requirements specified in this standard are based upon information provided by many sources within the Federal government and private industry. The requirements are designed to protect against adversaries mounting cost-effective attacks on unclassified government or commercial data. The primary goal in defining effective security for a system is to make the cost of any attack greater than the possible payoff.

While the security requirements specified in this standard are intended to maintain the security of a key recovery component, conformance to this standard does not guarantee that a particular component is secure. It is the responsibility of the manufacturer of a key recovery component to build the component in a secure manner.

Similarly, the use of a key recovery component that conforms to this standard in an overall system does not guarantee the security of the overall system. It is the responsibility of an organization operating a key recovery system to ensure that an overall system provides an acceptable level of security.

Since a standard of this nature must be flexible enough to adapt to advancements and innovations in key recovery technology, this standard will be initially reviewed in two years in order to consider new or revised requirements that may be needed to meet technological changes.

16. Waiver Procedure. To be supplied by the Federal Government.

17. Where to Obtain Copies of the Standard. To be supplied by the Federal Government.

Federal Information
Processing Standards Publication XXX

(Date)

Specifications for the

REQUIREMENTS FOR KEY RECOVERY PRODUCTS

1 Overview

Federal Agencies have a right and a responsibility to protect the information and data contained in, processed by, and transmitted between their IT systems. Ownership of the information is often shared with individuals, companies, and organizations and therefore requires that the government protect that information on its own behalf and on behalf of those co-owners. That protection must meet or exceed Federal Government standards and the standards of those co-owners.

Encryption is an important tool for protecting the confidentiality of communicated or stored data. When suitably strong encryption algorithms are employed and implemented with appropriate assurance, encryption can prevent the disclosure of communicated or stored data to unauthorized parties. However, the unavailability, loss, or corruption of the keys needed to decrypt encrypted data may prevent disclosure to authorized parties. To facilitate authorized access to encrypted data in the face of such failures, this Standard establishes requirements for key recovery products.

1.1 Scope of the Standard

This Standard neither requires nor endorses any specific technology for use in a Key Recovery System (KRS). It endeavors to be technology independent, so as not to unduly impede innovation in this new area. However, it is not the case that every conceivable key recovery technology will be amenable to successful evaluation under this Standard, e.g., intrinsically insecure KRS technologies may not be able to be evaluated.

This Standard presents a general model for a KRS. The model identifies functions that are intrinsic to any KRS: the generation of Key Recovery Information (KRI), the management of KRI, requests for key recovery, and the satisfaction of such requests by one or more Key Recovery Agents (KRAs). The Standard establishes functional, security, security assurance and interoperability requirements that apply to an implementation of each KRS function.

A product submitted for evaluation under this Standard must embody one or more of the KRS functions defined in this Standard. There is no requirement that a product offered for evaluation embody all of the defined functions; a compliant product may not constitute a complete KRS. There is no requirement that a single product or a suite of products from a single vendor embody all of the functions needed to provide a complete KRS. Thus, the Standard permits the modular implementation of a KRS, based on the assembly of products from one or more sources. Since an organization employing key recovery will require a complete KRS, additional guidance should be provided via other documents to assist in evaluating the security of a system assembled from products (from one or more vendors) that have been evaluated against this standard.

The security of a KRS is dependent on a mix of security disciplines, including computer, communication, procedural, physical, and personnel security. This Standard addresses only the computer and communication aspects of KRS security. Other critical aspects of KRS operation are outside the scope of this Standard. For example, a KRS must be available and survivable if it is to ensure authorized access to encrypted data, but this Standard does not address such concerns. Thus, compliance with this standard represents a set of necessary but not sufficient conditions for overall KRS security and utility.

For example, many key recovery schemes make use of public key technology and an associated public key infrastructure (PKI). The security of the resulting KRS is highly dependent on the security of the associated PKI. However, the many aspects of PKI security are outside the scope of this standard.

If key recovery is offered as a service by an organization ~~trusted third party~~, that ~~party~~ organization could employ products (e.g., a KRA) that comply with this Standard. However, the use of compliant products does not ensure the security for a KRS as a whole, nor does it ensure available or survivable KRS operations, as noted above. Hence, a KRS service cannot be said to comply with this Standard.

1.2 Road Map for the Standard

Section 2 of this Standard defines the abstract model for a KRS and defines the functions essential to KRS operation. Any product claiming compliance must identify which KRS functions are embodied in the product. Section 2 establishes functional and interoperability requirements for identified KRS functions. A product submitted for certification relative to this FIPS will be evaluated against the functional and interoperability requirements applicable to the functions that a vendor asserts are embodied in the product.

Section 3 defines the security requirements for KRS functions. Two levels of compliance are defined: Level 1 and Level 2. An implementation of a function at Level 1 provides basic security functionality, whereas Level 2 offers a higher level of security functionality. The choice of level for an application or environment is context sensitive, a function of many factors, and this Standard provides no guidance to prospective users in this regard. . However, any product claiming compliance with this Standard must declare the level at which each function of the product is asserted to comply (i.e., the level of compliance claimed by the developer). Because of the mapping between security levels and security assurance levels, it is not necessary to separately assert assurance level compliance.

Section 4 defines security assurance requirements for the implementation of KRS functions. These requirements are derived from the Common Criteria[2], and represent a profile of that security assurance evaluation criteria for use in this context. Three levels of (increasing) security

[2] SC 27 N1953, Evaluation Criteria for IT Security, Part 3 – Security Assurance Requirements.

assurance are defined: A, B and C. For each KRS function defined in Section 2, and each security functionality level defined in Section 3, one of these three assurance levels apply. Thus, there is a one-to-one correspondence between security functionality and assurance levels, on a per-function basis.

Appendix A provides illustrative examples based on the two key recovery schemes currently in use – encapsulation and escrow. Examples are provided for communication between two encapsulation schemes, between two escrow schemes, between an encapsulation and an escrow scheme, and between each of these schemes and a system with no key recovery.

Appendix ~~A~~ B contains illustrative examples of how to map the functions defined in the model in Section 2 to sample KRS products in the context of common applications. It also includes examples of how to map several existing key recovery system technologies to these functions. These examples are provided to assist vendors and evaluators in understanding the KRS functional model, but are not normative.

Appendix ~~B~~ C describes the concept of a Key Recovery Block (KRB), a data structure that would ~~format based on work (in progress) by the Key Recovery Alliance. The adoption of this format would~~ facilitate the encapsulation of KRI from different key recovery schemes and allow validation of the integrity of KRI in a KRS in support of requirements specified in Section 2. ~~However, the use of the KRB specification in this appendix is not a requirement of the Standard~~.

Appendix ~~C~~ D defines ~~an~~ two extensions for X.509 v3 certificates: one for use with a certificate associated with a KRA and one for use -with subscriber certificates in conjunction with certain private key escrow schemes.~~and a profile for other extensions employed in such certificates.~~ Many KRS designs make use of public key certificates. -The extensions defined here provide~~s~~ a standard means of representing certain data supportive of several KRS requirements. This appendix provides guidance for KRS designers and standards bodies who choose to make use of X.509 v3 certificates in support of key recovery, but this Standard does mandates neither the use of X.509 certificates nor of these extensions.

~~Appendix D contains illustrative examples of how key recovery enabled systems can be designed to maximize interoperability, both with systems that do not implement key recovery, and with systems that implement different key recovery schemes.~~

~~Appendix E provides an explanation of the two key recovery schemes currently in use – encapsulation and escrow. Examples are provided for communication between two encapsulation schemes, between two escrow schemes, between an encapsulation and an escrow scheme, and between each of these schemes and a system with no key recovery.~~

2 Key Recovery Model

A Key Recovery System (KRS) enables authorized persons to recover plaintext from encrypted data when the decryption key is not otherwise available. Key Recovery is a broad term that applies to many different key recovery techniques. Each technique will result in the recovery of a key – herein called the target key. The target key may be either:

- the data key that can be used to decrypt the data, or
- a key that can be used to decrypt the encrypted data key.

The information required to recover the target key may be different for each technique. The term “key recovery information” (KRI) will be used to refer to the aggregate of information needed by a key recovery technique to recover the target key. The key recovery information can be managed in a variety of ways. It may exist for only a brief time during electronic transmission, or it may exist for a relatively long time in storage. The KRI may be distributed among multiple location(s) (e.g., at one or more Key Recovery Agents (KRAs), ~~with a registration authority,~~ associated with or attached to a message or file, in end user systems, in third party systems, at a CA, in a certificate, or in a requestor facility).

Figure 1 presents a generalized model for a Key Recovery System, consisting of a KRI Generation, KRI Management and Key Recovery. The model addresses the creation of KRI for the recovery of the target key, the management of the KRI, and the recovery of the target key from that KRI.

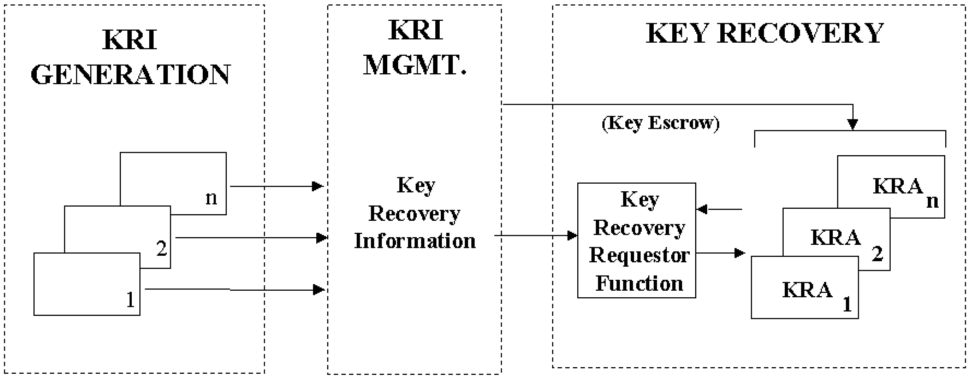


Figure 1: General Model for Key Recovery Systems

KRI generation is performed by a KRI Generation Function. KRI Management is performed by a KRI Delivery Function and a KRI Validation Function. Key Recovery is performed by a Key Recovery Requestor Function and a KRA Function. The resulting five functions are shown in Figure 2.

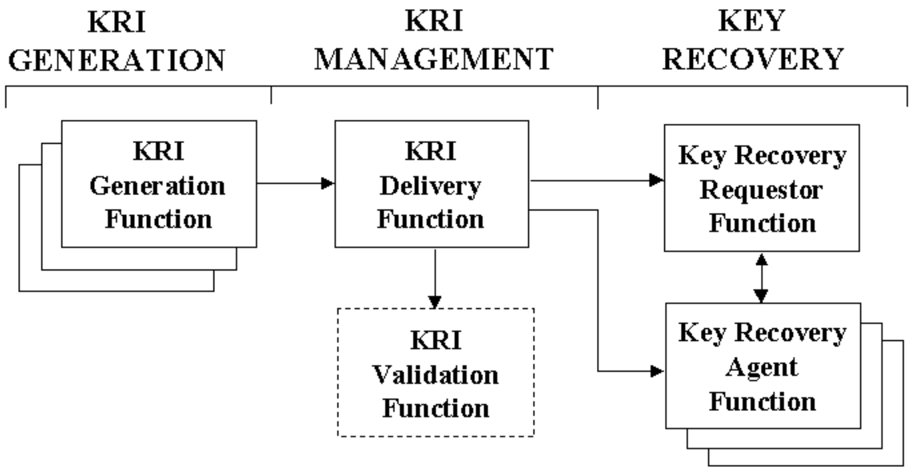


Figure 2: The Five Functions of a Key Recovery System

The key recovery model addresses multiple key recovery techniques (see Section 2.8) and supports a wide variety of data applications, including:

- Interactive communication sessions
- Store-and-forward communications
- Data storage

A Key Recovery System (KRS) may exist over multiple "locations" (e.g., cryptographic end systems, KRA systems, requestor system, and storage or transmission media). The normal key used by a target application exchange mechanism need not be affected by the use of key recovery mechanisms. However, key exchange mechanisms may be used to support the creation and distribution of key recovery information (e.g., the integration of KRI into existing key exchange mechanisms is not precluded). In the future, key exchange protocol designers may find it beneficial to integrate key recovery into the base design of the protocol.

Appendix A provides examples of the distribution of functions of the model within products implementing a Key Recovery System.

The functions of the Key Recovery Model specified in this standard must be implemented in products which, when used together with a key recovery policy and procedures, form a Key Recovery System. A key recovery policy specifies the conditions under which key recovery information must be created and the conditions under which key recovery information may be released. The policy identifies the authorized key requestors and specifies the conditions under which each requestor is authorized to access data. The policy may also indicate the allowable Key Recovery Agent(s), how or where key recovery information must be maintained, and whether or not the received encrypted information should be processed when key recovery information is not available. The key recovery policy could be "hardwired" (e.g., implemented in a manner which does not allow key recovery to be bypassed), selectable by a user, or implemented in policy management tables or modules.

The remainder of this section identifies functional and interoperability requirements for key recovery products which are designed to be conformant with this standard. Requirements are designated by "Req" numbers, and the requirement and its number are presented in a bold font. Explanatory text is provided in subsequent paragraphs.

(Req. 1) There shall be a well-defined mapping from the key recovery functions of a product to the functions of the key recovery model. A vendor shall provide a document describing the complete KRS scheme in which the product(s) submitted for evaluation are intended to operate. It shall be possible to test the described interfaces between the product(s) and the functions needed to provide a complete KRS scheme. ~~[ADD TEXT RE DETERMINING IF THE SUBMITTED PRODUCT IS A CRYPTOGRAPHIC END SYSTEM]~~

A product claiming compliance with the Standard must be mappable to one or more of the KRS functions defined in this Standard. There is no requirement that a product offered for evaluation embody all of the defined functions, nor is there a requirement that a single vendor provide a complete KRS. The modular implementation of a KRS, based on the assembly of components from one or more sources, is allowed. However, a vendor submitting a product for evaluation must provide a thorough description of how the KRS functions in the product fit into a complete KRS. The description must include all interfaces between the KRS functions embodied in the submitted product and any KRS functions with which these functions interact. For product evaluation, it must be possible to test these interactions, either by assembling a complete KRS, or through the use of simulation, test fixtures, or through analytic means. [text from Jan re ToO relationship]

(Req. 2) A vendor submitting a product for evaluation shall submit a theory of compliance document that describes how the product complies with all of the applicable requirements in this FIPS.

The scope of the theory of compliance document includes all of the requirements established in this FIPS, including functional, security, and assurance requirements. (A document addressing the security and assurance requirements is sometimes referred to as a "security target.")

~~(Req. 2)~~(Req. 3) **A product submitted for evaluation shall be configurable so that it would be possible to interoperate with some product(s) (extant or not) to form a complete KRS composed only from compliant KRS functions. Each KRS function in the selected subset shall be capable of operating independently of the functions outside of the selected subset.**

A product may be submitted for the evaluation of a subset of the KRS functions it provides. This allows a product to offer both compliant and non-compliant KRS functions, and receive certification only for the compliant functions.

~~(Req. 3)~~(Req. 4) **If a function in a product submitted for evaluation may operate in both compliant and non-compliant modes, the product shall be configurable so that one can determine unambiguously whether the compliant or non-compliant mode of the function will be invoked.**

2.1 Key Recovery Information (KRI) Generation Function

~~(Req. 4)~~(Req. 5) **Each instance of the KRI Generation Function shall generate all or part of the KRI. If KRI is generated by more than one instance of this function, the set of all KRI generating functions shall yield KRI sufficient for key recovery.**

The KRI Generation Function consists of one or more KRI-generating entities~~, also called KRI providers~~. A KRI~~- provider~~ generating entity could, for example, be the sender or receiver of a communication, a Certification Authority (CA), a Key Distribution Center, ~~a Registration Authority,~~ or a component vendor. The KRI may include the identity of a KRA, the identity of a key, a date and time, authorization information, an indication of the key recovery type and manufacturer, an algorithm identifier, an encrypted key, or pointer information (e.g., information that points to the location or holder of a key). The method in which this function is implemented often differs among key recovery schemes, hence no detailed requirements are expressed for this function.

The KRI Generation Function may be distributed over multiple locations (e.g., systems, or hardware or software products) - all KRI required to recover a given data key/ciphertext set need not be created by the same generating entity. For example, the entity generating an encryption key pair may be different than the entity using that key pair to secure the data key which was used to encrypt the ciphertext data. See Appendix A for further examples.

During an initialization or configuration stage, and at times of periodic updates, the KRI-generating entities obtain initialization information and cryptographic parameters, or otherwise are configured to establish shared information as necessary with the KRA(s) to allow key recovery. For example, for ~~KRI~~ key encapsulation systems ~~(see Appendix E)~~, initialization may involve obtaining authentic copies of the KRA public key(s) for subsequent use in encapsulating the KRI by the cryptographic end system. For key escrow systems ~~(see Appendix E)~~, initialization and configuration may involve setting parameters that will allow a secure communication channel to be established between a cryptographic end system and a KRA for the escrowing of private keys. These are critical aspects of the overall Key Recovery System, but their definition is beyond the scope of this document. [this is what we recently called RRI. Do we need to create one or more new functions for the model (Recovery Registration Information Generation, Delivery, …), update figures 1 and 2, add a new sub-section here in section 2, and corresponding sub-sections in section 3, plus new table entries in section 4.]

~~(Req. 5)~~(Req. 6) An instance of the KRI Generation Function assembles and formats all or part of the KRI for use by other key recovery functions.

The KRI Generation Function generates, assembles and formats the KRI, as appropriate, for consumption by the KRI Validation Function, the Key Recovery Requestor Function and the KRA Function. The format of the KRI and its delivery method is generally specific to a key recovery technique. Information may be acquired from multiple sources (e.g., one or more CA certificates, a key generation device or a time stamping device) in order to generate the required KRI necessary for a given key recovery technique.

A method is required for associating encrypted data with the KRI that can be used to recover that data. This may be accomplished in a product by (1) providing plaintext information pointing to the KRI within a structure containing the encrypted data, (2) providing plaintext information pointing to the encrypted data within a structure containing the KRI, (3) by a well-defined placement of the KRI and the encrypted data (e.g., within the same message), (4) by acquiring information from another source associated with the encrypted data (e.g., by examining a certificate to determine that a key is escrowed), or (5) by a combination of such techniques.

~~(Req. 6)~~(Req. 7) The KRI Generation Function is responsible for ensuring the validity of its output.

This includes all information generated by the function itself, as well as information generated by other sources (e.g., another KRI Generation Function, a CA, time stamping authority, etc.) which are used in the assembly and format process. In some instances this requirement may be met by authenticating the sources of inputs to KRI generation, as opposed to validating the inputs themselves.

~~(Req. 7)~~(Req. 8) The KRI Generation Function shall provide the generated KRI to the KRI Delivery Function.

~~(Req. 8)~~(Req. 9) **A Level 2 product shall not provide a facility to deactivate KRI generation.**

For a Level 1 product, KRI generation may be configurable. In a Level 2 product, there must be no facility to deactivate KRI generation.

2.2 KRI Delivery Function

The KRI Delivery Function makes the generated KRI available for validation and recovery (e.g., by storing or transmitting the KRI). The KRI Delivery Function may be distributed over multiple locations (e.g., systems, or hardware or software products).

~~(Req. 9)~~(Req. 10) **When KRI is delivered in conjunction with a standard communication protocol, the transmission format shall be determined by that protocol standard.**

There are a number of standard communication protocols that allow the use of encryption to protect the data carried by that protocol. When KRI is introduced into one of these communication protocols, it must be done in a manner that preserves the ability to communicate (see Section 2.7, Interoperability).

~~(Req. 10)~~(Req. 11) **The KRI Delivery Function shall store KRI with persistence and availability commensurate with that of the corresponding stored ciphertext.**

KRI for a given data key/ciphertext pair must be available for the duration of time that the given ciphertext exists. If the ciphertext is decrypted and subsequently not available in its original ciphertext form (e.g., stored in plaintext or re-encrypted with a different data key), then the original KRI is no longer required. The KRI Delivery Function is expected to call upon normally available storage system resources to effect appropriate persistence and availability, but no extraordinary measures need be employed.

~~(Req. 11)~~(Req. 12) **The KRI Delivery Function shall make the KRI available to the Key Recovery Requestor Function or the KRA Function or both.**

The KRI Delivery Function shall make the KRI available to the Key Recovery Requestor Function or the KRA Function(s) or a combination thereof. The term “make available” is system dependent and includes sending the KRI to the Key Recovery Requestor directly, or depositing the KRI in one or more locations known to and accessible by the Key Recovery Requestor (i.e., the requestor(s)).

~~(Req. 12)~~(Req. 13) The KRI Delivery Function (for level 2 compliance) shall make the KRI available to the KRI Validation Function.

The KRI Delivery Function must provide the KRI produced by the KRI Generation Function to the KRI Validation Function. The method of delivery may be via a communication channel, storage device or directly between modules within the same system.

2.3 KRI Validation Function

~~(Req. 13)~~(Req. 14) For level 2 compliance, if KRI Validation fails, access to plaintext at the cryptographic end system shall be denied.

The KRI Validation Function ensures that KRI is valid and usable for key recovery. The intent of this function is to provide assurance that a key requestor can use KRI to successfully recover a target key in order to recover encrypted data. Several methods of validation may be performed, including:

- Checking certificates for the presence of KRI (e.g., KRA identities, key recovery technique),
- Checking that KRI is available for a KRA (e.g., in a recipient list or a key recovery block),
- Authenticating the source of the KRI,
- Validating the integrity of KRI associated with the encrypted data (e.g., received in the same message), and
- Verifying that the KRI can actually be used to recover the data key needed to decrypt the encrypted data (e.g., the correct target key can be produced).
- Creating KRI, either when no KRI is received or in lieu of accepting and verifying KRI that is received, or if validation of received KRI is not successful. (In the last example, failure of the received validation is “overridden” by the receiver’s generation of KRI.) In this case, a KRI Generation Function must be available.

2.4 Key Recovery Requestor Function

The Key Recovery Requestor Function authenticates the entity making the request to the Key Recovery Agent. The Key Recovery Requestor Function consists of the requestor and a

Requestor Subsystem (see Figure 3). The requestor is an entity who seeks to recover information that will allow the decryption of encrypted data. A request for~~ a~~ key recovery~~ service~~, made by a requestor using a Requestor Subsystem to interact with one or more Key Recovery Agents, must be an authorized request -- the requestor and the Requestor Subsystem that issues a request for ~~a~~ key recovery ~~service~~ must be authorized under system policy to access the data that can be decrypted using the recovered target key. Furthermore, the requestor and the Requestor Subsystem must establish their right to access that data. The authentication and authorization process is beyond the scope of this standard.

~~(Req. 14)~~(Req. 15) For given KRI, the ~~Key Recovery Requestor~~KRR Function shall have the ability to recover a target key by interacting with one or more Key Recovery Agents.

The requestor provides key recovery information to the Requestor Subsystem. The Requestor Subsystem interacts with one or more KRAs to obtain either ~~the~~ a target key, or multiple key parts or key related information which will allow the reconstruction of ~~the~~ a target key. The target key may then be used to recover the data using a Data Recovery System. The Data Recovery System is not specified in this standard.

KRI may be designed so that one KRA may not be able to provide all the information necessary to recover a target key. For example, each KRA may be able to provide key products which are then combined to reconstruct the target key.

(Req. 16) Encrypted data transmitted by the KRR Function shall be recoverable.

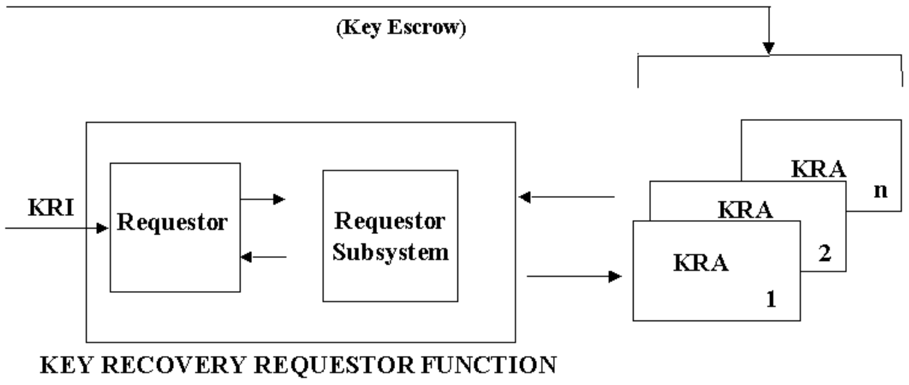


Figure 3: Key Recovery Functions

This requ nication is recoverable.

2.5 Key Recovery Agent Function~~(s)~~

A Key Recovery Agent (KRA) Function, is a trusted function that performs~~ a~~ key recovery ~~service ~~in response to an authorized request made by a Requestor Subsystem on behalf of a requestor.

~~(Req. 15)~~(Req. 17) The KRA shall store keys, key components or any other information required to satisfy the recovery of a target key .

(Req. 18) All of the data needed to operate the KRA, and all cryptomodules employed by the KRA, must be securely replicable, in support of availability.

Provision of a facility to duplicate the databases and to instantiate duplicate (equivalent) cryptomodules satisfies this requirement. There is not a requirement for the replicated KRA to be available online; use of an archive capability satisfies this requirement so long as the KRA can be reconstituted from the backup database and through use of a distinct (but equivalent) cryptomodule.

~~(Req. 16)~~(Req. 19) A Key Recovery Agent Function shall have the ability to process the KRI provided by the Key Recovery Requestor Function. Processing by the Key Recovery Agent Function shall yield some or all of the information required to decrypt data acquired by a Requestor.

The key recovery service performed by a KRA consists of processing all or part of the KRI provided to the KRA by the Requestor Subsystem, and returning an output value to the Requestor Subsystem. The output value may be either the target key, or multiple key parts or key related information which will allow the reconstruction of the target key.

(Req. 20) Encrypted data transmitted by the KRA Function shall be recoverable.

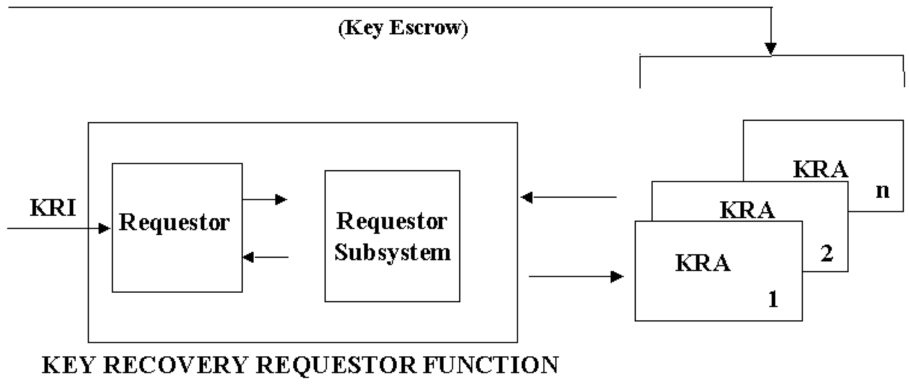


Figure 44~~3~~: Key Recovery Functions

This requirement, and its complement in the preceding section, ensure that KRR-KRA communication is recoverable.

2.6 Cryptographic End Systems

The functions of the Key Recovery Model specified in this standard must be implemented in products which, when used together with a key recovery policy and procedures, form a Key Recovery System. The key recovery functions within the model may be distributed across these products as appropriate for the specific key recovery technique and the key recovery policy adopted for an organization. This section defines the concept of a cryptographic end system, as needed to support validation of interoperability requirements.

~~(Req. 17)~~(Req. 21) A vendor submitting a product for evaluation under this Standard shall declare the product as a cryptographic end system if it encrypts or decrypts application data using a target key and incorporates a KRI Generation, KRI Delivery, or KRI Validation Function.

In order to recover encrypted data, the key recovery information must be generated in order to allow the recovery of data keys used by that system. The KRI may be made available in various ways, e.g., as encapsulated information which may be stored or communicated with the encrypted data, or as escrowed data, or both.

The model does not specify which system or systems generate the KRI. When KRI is generated by cryptographic end systems, the KRI could be generated by the entity that encrypts data (e.g., the sender) or the entity that decrypts data (e.g., the receiver). A cryptographic end system generates and processes KRI in accordance with a specified key recovery policy.

Note that cryptographic end system products need not contain a specific set of key recovery functions (see Appendix A). The use of the functions within a cryptographic end system can depend on which key recovery technique is being used and whether the system is acting as a sender or receiver system. When a key encapsulation application is acting as a sender, it would typically perform the KRI Generate and Delivery Functions, whereas when acting as a receiver, it would often perform the KRI Validation Function. In a key escrow-based application, however, the sender may perform the KRI Validation Function, rather than the receiver.

2.7 Interoperability

[steve to re-write]

This standard establishes interoperability requirements for ~~several types of key recovery system products:~~ cryptographic end systems~~, Key Recovery Agents and Key Recovery Requestors~~. No interoperability requirements are imposed on communication between a cryptographic end system and a Key Recovery Agent (KRA) or between a KRR and a KRA. In this latter case, the imposition of interoperability requirements is viewed as potentially too restrictive in light of the wide range of key recovery technologies that this Standard attempts to embrace.

This standard will define a syntax for communication between a Key Recovery Requestor (KRR) and a KRA. This syntax applies only to electronic key recovery transactions effected via a communication medium (e.g., telephone, LAN or Internet). Key recovery transactions effected via storage media (e.g., diskette or tape) or via direct interaction (e.g., self recovery on a PC) are not covered by these requirements. These syntactic requirements have been established to reduce life cycle costs for users of key recovery systems and because it appears to be feasible to do so without introducing undue constraints on technology options. Section 5 defines the syntax for this communication. No interoperability requirements are imposed on communication among KRAs from different vendors.

Interoperability requirements for cryptographic end systems apply only to the use of key recovery for communicated data, not for data storage. With regard to such systems, interoperability requirements apply only in the context of systems that communicate in an interoperable, encrypted fashion, exclusive of the use of key recovery technology. Such systems fall into two categories: those that make use of "standard" communication protocols and those that make use of "proprietary" protocols. For this standard, the phrase "standard communication protocol" encompasses any communication protocol that has been adopted by a generally-recognized protocol standards organization, including the International Telecommunication Union (ITU), International Organization for Standardization (ISO), the American National Standards Institute (ANSI), the Institute of Electrical and Electronics Engineers (IEEE), the Asynchronous Transfer Mode (ATM) Forum and the Internet Engineering Task Force (IETF).

No interoperability requirements are established for cryptographic end systems that engage in encrypted communications using proprietary communication protocols. Such systems typically exhibit limited interoperability (except within individual vendor product lines) due to the use of non-standard protocols. Still, vendors who choose to incorporate key recovery technology in their products are encouraged to do so in a fashion that minimizes disruption to the installed product base in order to facilitate communication between key recovery products and non-key recovery products.

~~(Req. 18)~~(Req. 22) The cryptographic end system shall be configurable so that interoperability is preserved when communicating with key recovery capable or non-key recovery capable end systems.

When key recovery is introduced into a system using a standard (encrypted) communication protocol, it must be done in a fashion that preserves interoperability, i.e., if two systems were able to communicate securely prior to the introduction of key recovery technology, then they must be able to do so after the introduction of the technology. Some key recovery capable systems may be configured so that they will refuse to communicate with other systems unless it can be determined that the other systems are employing key recovery. If this feature is activated, it may prevent interoperability between otherwise interoperable systems. However, the presence of this configurable feature does not exempt a system from meeting the interoperability requirements detailed below. There are two general approaches to meeting this requirement.

If a key escrow scheme ~~(see Appendix E)~~ is employed, the (extant) secure communication protocol employed by the cryptographic end systems need not be modified to carry any key recovery information, and thus, interoperability is inherently preserved. Note that in this case, interoperability is preserved both among key recovery capable systems, and between key recovery capable and non-key recovery capable systems. If no changes are made to the secure communication protocol, including any supporting key and/or certificate management protocols, then it may or may not be possible for communicating systems to determine if key recovery is being employed. If a key escrow scheme elects to transmit some information in a secure communication protocol to indicate that key recovery is enabled, then it must do so in a fashion that does not impair interoperability. For example, if X.509 public key certificates are employed to support secure communication, an extension can be added to each certificate specifying the KRA(s) for the subject. If such an extension is employed and not marked “critical”, this approach complies with the interoperability requirement established here. However, if such an extension were employed and marked “critical”, this would not be compliant, as it would inhibit interoperability with non-key recovery aware systems. See Appendix C for a proposed X.509 certificate extension.

If a ~~KRI~~ key encapsulation scheme ~~(see Appendix E)~~ is employed, the key recovery information will be carried in the secure communication protocol. In some standard, secure communication protocols, it is possible to carry this information in a fashion that preserves interoperability without modifying the protocol. For example, in a secure e-mail protocol (e.g., MSP[3], PGP[4], S/MIME[5], or X.411[6] an additional recipient, representing a KRA, could be added to the per-recipient token list to provide key recovery on a per message basis.

[3] Message Security Protocol (MSP), Specification SDN.701 Revision 3.0 1994-03-21

[4] REFERENCE NEEDED

[5] Secure Multipurpose Internet Mail Extension

[6] ITU-T: Information technology - Message Handling Systems (MHS): Message transfer system: Abstract service definition and procedures,11/1995

In a session key management protocol, one party may transmit per-session KRI. For example, the IEEE 802.10c Key Management protocol[7] incorporates an optional field in the Pick-SA-Attrs exchange to carry KRI. In ISAKMP[8], one party can transmit a (yet to be defined) NOTIFY message with a payload containing per-session KRI. A compliant ISAKMP implementation will silently discard an unrecognized payload, thus preserving interoperability. These approaches to key recovery are compliant with the interoperability requirements established in this Standard.

If it is necessary to transport KRI, and there is no provision in a standard communication protocol for doing so in an interoperable fashion, then it will be necessary to modify/extend the protocol to carry such information. It is outside the scope of this standard to specify how key recovery information should be transported in the context of such protocols. The definition of an interoperable means of carrying such information is solely the purview of the cognizant standards body for each affected protocol.

~~(Req. 19)~~(Req. 23) A vendor of a cryptographic end system shall provide documentation demonstrating that the product transports KRI in a fashion consistent with the specification developed and adopted by the cognizant standards body for the protocol in question.

[7] IEEE 802.10c/D6, Standard for Interoperable LAN Security-Part C: Key Management.

[8] Internet Security Association Key Management Protocol

3 Security Requirements

This section defines security requirements for all of the functions defined in the KRS model established in Section 2. The security requirements have been defined to allow a variety of product architectures. These include using a monolithic product on which no other software/firmware can be loaded, using a monolithic product on which other software/firmware can be loaded, or using a layered product that has a distinct operating system, application, and cryptographic module.

The requirements for the KRA and the Key Recovery Requestor Functions have been defined so that all of these architectures can be evaluated. This is especially true of the requirements in the following areas: Audit, Identification and Authentication, Access Control, and Protection of Trusted Security Functions.

Furthermore, a product architecture may imply that some of the requirements do not apply, e.g., a requirement intended to mitigate a threat that does not arise in a particular implementation model. For example, if the product is a monolithic product on which no other software/firmware can be loaded, the domain separation, trusted path, and reference validation mechanism requirements do not apply since the untrusted software threat does not exist.
[check to see if introduction of self recovery notion, e.g., level 0 KRR, interactions badly with the following KRA requirements.]

3.1 Key Recovery Agent Function Requirements

3.1.1 Level 1 – Medium Assurance

3.1.1.1 Cryptographic Functions

~~(Req. 20)~~(Req. 24) All cryptographic modules shall be compliant with FIPS 140-1, Level 2 or higher.

3.1.1.2 Cryptographic Algorithms

~~(Req. 21)~~(Req. 25) A KRA function submitted for evaluation shall be able to be configured to use only FIPS approved algorithms (where applicable).

If a cryptographic function can be effected using a FIPS approved algorithm, it must be possible to configure the KRA to make use of this algorithm. However, if a key recovery scheme requires a cryptographic function not supported by any FIPS approved algorithms, there is no requirement to make use of such algorithm, e.g., use of RSA[9]for key encapsulation.

3.1.1.3 Confidentiality

These requirements are intended to protect against both outsider and insider threats. The only insider threat addressed is the unauthorized user. The authorized insider threat is handled elsewhere using audit, role separation, and multi-person control.

~~(Req. 22)~~(Req. 26) The KRA Function shall protect all stored sensitive data (e.g., KRI, TKI, [and RRI ?]) against disclosure to unauthorized individuals.

This requirement also applies to copies of sensitive KRA data retained in backup/archive form, in support of Requirement <reference to new 16>.

~~(Req. 23)~~(Req. 27) The KRA Function shall protect target key information transmitted - electronically or physically communicated - against disclosure to unauthorized individuals.

~~(Req. 24)~~(Req. 28) The strength of the encryption algorithm used to protect target key information shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for the generation of the keys being recovered.

(Evaluation guidance documents will provide details on how to compare encryption algorithms in support of this requirement.)

~~(Req. 25)~~(Req. 29) The product shall apply confidentiality services to all outgoing transactions. The strength of the algorithm used for confidentiality shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.

3.1.1.4 Integrity

~~(Req. 26)~~(Req. 30) The product shall protect all stored KRI [and RRI ?]against modification.

[9] ANSI X9.31, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)

~~(Req. 27)~~(Req. 31) **The product shall apply data origin authentication to all outgoing transactions (i.e., requests and responses). The strength of the algorithm used for authentication shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption and for generation of the keys being recovered.**

~~(Req. 28)~~(Req. 32) **The product shall apply data integrity services to all outgoing transactions. The strength of the algorithm used for integrity shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.**

3.1.1.5 Audit

These requirements are used to create a log of information to allow oversight by a security officer to detect unauthorized operations by a Key Recovery Agent. The recording of events defined as "auditable" may be enabled under configuration control.

~~(Req. 29)~~(Req. 33) **The KRA shall cease operation if it is unable to effect audit operations.**

~~(Req. 30)~~(Req. 34) **The product shall generate an alarm to ~~the~~ an authorized administrator if the size of the audit data in the audit trail exceeds a pre-defined limit.**

~~(Req. 31)~~(Req. 35) **The product shall provide ~~the~~ an authorized administrator with the ability to manage the audit trail at any time during the operation of the product.**

~~(Req. 32)~~(Req. 36) **Keys shall not be included in audit trails.**

~~(Req. 33)~~(Req. 37) **The following events shall be auditable:**

(a) Any specific operation performed to process audit data stored in the audit trail (Note: This includes emptying, backup and deletion of audit trail);
(b) Any attempt to read, modify or destroy the audit trail;
(c) All requests to use authentication data management mechanisms;

(d) All modifications to the audit configuration that occur while the audit collection functions are operating;
(e) All requests to access user authentication data;
(f) Any use of an authentication mechanism. (e.g. login);
(g) All attempts to use the user identification mechanism, including the user identity provided;
(h) Use of a security-relevant administrative function;
(i) Explicit requests to assume ~~the~~ a security administrative role;
(j) The allocation of a function to a security administrative role;
(k) The addition or deletion of a user to/from a security administrative role;
(l) The association of a security-relevant administrative function with a role;
(m) The invocation of the non-repudiation service. The audit event shall include the identification of the information, the destination, and a copy of the evidence provided. The event shall exclude all private and secret keys in encrypted or unencrypted form.
(n) All attempted uses of the trusted path functions; and
(o) Identification of the initiator and target of the trusted path.

~~(Req. 34)~~(Req. 38) It shall not be possible to disable the auditing of an event defined as “always audited.”~~The recording of an event defined as “always audited” shall not be disable-able.~~

~~(Req. 35)~~(Req. 39) The following events shall always be audited.
(a) Requests, responses, and other transactions received by the product, including key recovery requests;
(b) Requests, responses, and other transactions generated by the product, including key recovery responses;
(c) Start-up and shutdown of the audit functions.

~~(Req. 36)~~(Req. 40) The product shall record at least the following information within each audit record:
(a) Date and time of the event, type of event, subject (user) identity, and success or failure of the event;
(b) Other audit event type information as follows:
(1) For changes to the configuration file event, changes shall also be recorded in the audit record.
(2) When attempting a function using ~~the~~ a security administrative role, the function attempted, the role and all applicable inputs shall be recorded in the audit record.

(3) When allocating a function to a security administrative role, the role and the function shall be included in the audit record.

(4) When adding or deleting users to/from ~~the~~ a security administrative role, the role, user identity and the addition/deletion action shall be included in the audit record.

(5) For all ~~KRA~~ transactions, the entire transaction (excluding keys and TKI) shall be included in the audit record as sent or received.

~~(Req. 37)~~(Req. 41) The product shall be able to generate a human understandable presentation of any audit data.

~~(Req. 38)~~(Req. 42) The audit trail shall not store old or new authentication information (e.g., passwords).

~~(Req. 39)~~(Req. 43) The product shall be able to associate each auditable event with the identity of the user that caused the event.

~~(Req. 40)~~(Req. 44) The product shall provide ~~the~~ an authorized administrator with the ability to empty the audit trail.

Note: emptying the audit trail means backup and delete.

~~(Req. 41)~~(Req. 45) The product shall be able to include or exclude auditable events from the set of audited events based on the following attributes: User identity, and/or Event Type.

~~(Req. 42)~~(Req. 46) The product shall restrict access to the audit trail to ~~the~~ an authorized administrator.

3.1.1.6 Identification and Authentication

These requirements support the unique identification of KRA personnel. This facilitates individual accountability via audit functions and access controls. Requirements are levied on the strength of the authentication mechanism against attacks by rogue KRA personnel.

These requirements do not apply to electronic transactions (requests and responses). The electronic transactions may be identified and authenticated (if the scheme permits) using the access control policy.

Note: If ~~the~~ a crypto officer is invoking a KRA cryptographic module function, authentication may be effected directly to the module and is exempt from all of the ~~following newly added~~ requirements of this section. In this case, the FIPS 140-1 level 2 module I&A requirements apply.

~~(Req. 43)~~(Req. 47) The product shall provide functions for initializing and modifying KRA personnel authentication data.

~~(Req. 44)~~(Req. 48) The product shall restrict the use of initialization and modification of the KRA personnel authentication data to ~~a~~ security administrators.

~~(Req. 45)~~(Req. 49) The product shall allow authorized KRA personnel to modify their own authentication data.

~~(Req. 46)~~(Req. 50) The product shall protect authentication data that is stored in the product from unauthorized observation, modification, and destruction.

~~(Req. 47)~~(Req. 51) The product shall protect authentication information from unauthorized reuse, including replay.

Note: This requirement and the previous requirement provide a capability for secure remote login.

~~(Req. 48)~~(Req. 52) The product shall be able to terminate the ~~KRA personnel~~ session establishment process after at most five consecutive unsuccessful authentication attempts.

~~(Req. 49)~~(Req. 53) After the termination of ~~a~~ the ~~KRA user~~ session establishment process, the product shall be able to disable the user account until the account is enabled by a security administrator .

~~(Req. 50)~~(Req. 54) The product shall authenticate ~~any~~ the ~~KRA operator's~~ claimed identity of an individual prior to performing any functions on ~~that operator's~~ the behalf of that individual.

~~(Req. 51)The product shall authenticate each KRA operator before performing any actions requested by that operator.~~

~~(Req. 52)~~(Req. 55) The product shall require a user authentication technology that protects authentication information capture (this requirement is met by a trusted path or the use of a one time password). The strength of the mechanism shall nominally reduce the likelihood of false authentication to less than 1/1,000,000~~in terms of space shall meet the requirement of 1 in 1,000,000~~.

Techniques that meet this requirement are defined in FIPS PUB 112 based passwords entered via a trusted path, RFC 1938 (One Time Password), hardware tokens connected via trusted channels/paths, and biometric tokens connected via trusted channels/paths.

~~(Req. 53)~~(Req. 56) If the product makes use of a "trusted path" mechanism to meet the preceding I&A requirement, that trusted path between itself and ~~local human users~~KRA personnel shall be logically distinct from other communication paths and shall provide an assured identification of its endpoints. ~~The local human user~~KRA personnel shall have the ability to initiate communication via this trusted path.

3.1.1.7 Access Control

These requirements provide countermeasures against an entity masquerading as an authorized requestor or KRI generator. The requirements in this section address the security of electronic communication between the KRA and the Requestor Subsystem or KRI Generation Function. If these interactions are not electronic, then physical and procedural means must be used to secure the transactions. These procedural and physical measures are beyond the scope the Standard.

~~(Req. 54)~~(Req. 57) The product shall unambiguously associate a received response to an outstanding request. The strength of the algorithm used for the association shall be greater than or equal to the strength of the encryption and key management algorithms employed for the encryption of user traffic or for the generation of the keys being recovered.

~~(Req. 55)~~(Req. 58) The product shall release target key information only to authorized requestors.

~~(Req. 56)~~(Req. 59) The product shall release target key information only if the requestor is authorized to receive the data associated with the KRI and for the validity period (time interval) specified in the request, and

only if any additional conditions for release (specified in the KRS policy) have been satisfied .

KRA products are not required to support additional conditions for release as a prerequisite for evaluation.

~~[NOTE THAT THE KEY RECOVERY REQUESTOR-KRA SYNTAX SHOULD INDICATE WHETHER THE VALIDITY PERIOD IS OPTIONAL. IF IT IS ALWAYS PRESENT, THE KRA MAY IGNORE IT IF THE KEY RECOVERY SCHEME DOES NOT ALLOW FOR TIME INTERVAL-BASED TARGET KEY INFORMATION RELEASE.]~~

~~(Req. 57)~~(Req. 60) The product shall ensure that security features are always invoked and cannot be bypassed.

~~(Req. 58)~~(Req. 61) The product shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

~~(Req. 59)~~(Req. 62) The product shall enforce separation between the security domains of subjects in the system.

~~(Req. 60)~~(Req. 63) The product shall restrict the ability to perform security-relevant administrative functions to a security administrative role that has a specific set of authorized functions and responsibilities.

Note: The term “security administrative role” refers to generic trusted administrative roles. The system administrator role is one, but not the only one, of these security administrative roles. Additional security administrative roles are defined later in Requirement (Req. 84)~~(Req. 83)~~~~(Req. 81)~~.

In order to meet the preceding requirements, the product must distinguish security-relevant administrative functions from other administrative functions. The set of security-relevant administrative functions must include all functions necessary to install, configure, and manage the product; minimally, this set must include:

- the assignment/deletion of authorized users from security administrative roles,
- the association of security-relevant administrative commands with security administrative roles,
- the assignment/deletion of subscribers~~subjects~~ whose keys are held,
- the assignment/deletion of parties who may be provided the keys,
- product cryptographic key management,
- actions on the audit log, audit profile management, and
- changes to the system configuration.

~~(Req. 61)~~(Req. 64) The product shall be capable of distinguishing the set of KRA personnel authorized for administrative functions from ~~the set of~~ all other ~~users~~personnel.

~~(Req. 62)~~(Req. 65) The product shall allow only specifically authorized KRA personnel to assume ~~the~~ a security administrative role.

~~(Req. 63)~~(Req. 66) The product shall require an explicit request to be made in order for an authorized KRA operator to assume ~~the~~ a security administrative role.

3.1.1.8 Authentication of Received Transactions

~~(Req. 64)~~(Req. 67) The product shall verify the source of received transactions.

~~(Req. 65)~~(Req. 68) The product shall verify the integrity of received transactions.

3.1.1.9 Non-Repudiation

These capabilities facilitate the use of a trusted time source to further support accountability.

~~(Req. 66)~~(Req. 69) The product shall provide trusted time stamps for use in transactions with requestors.

~~(Req. 67)~~(Req. 70) The product shall generate evidence of origin for transmitted key recovery responses.

~~(Req. 68)~~(Req. 71) If the product receivesKRI [RRI?], ~~T~~the product shall generate evidence of receipt for ~~the registration of target key information.~~it.

~~(Req. 69)~~(Req. 72) The product shall verify evidence of origin for key recovery requests and for ~~target key information registration~~ KRI [RRI?] transactions.

3.1.1.10 Protection of Trusted Security Functions

~~(Req. 70)~~(Req. 73) Before establishing a session with a KRA administrator, the product shall display an advisory warning message regarding unauthorized use of the product.

~~(Req. 71)~~(Req. 74) The default advisory warning message displayed by the product shall be as follows: “This system shall be used only by authorized personnel and only for authorized key recovery purposes. Violation may result in criminal prosecution and civil penalties”.

~~(Req. 72)~~(Req. 75) The product shall restrict the capability to modify the warning message to ~~the~~ an authorized security administrative~~or~~ role.

~~(Req. 73)~~(Req. 76) Upon successful session establishment, the product shall display the date, time, method, and source of the last successful session establishment to the KRA operator.

~~(Req. 74)~~(Req. 77) Upon successful session establishment, if there have been any unsuccessful session establishment attempts since the last successful session establishment, the product shall display the date, time, method, and location of the most recent unsuccessful attempt to establish a session as well as the number of unsuccessful attempts since the last successful session establishment.

~~(Req. 75)~~(Req. 78) The data specified above shall not be removed without KRA operator intervention.

3.1.2 Level 2 – High Assurance

3.1.2.1 Cryptographic Functions

~~(Req. 76)~~(Req. 79) KRA cryptographic modules shall be compliant with FIPS 140-1, Level 3 or higher.

Note: This requirement does not apply to cryptographic modules used for KRA administrator I&A.

3.1.2.2 Cryptographic Algorithms

Same as Level 1.

3.1.2.3 Confidentiality

Level 2 requires additional protection against the insider threat of a rogue Key Recovery Agent by requiring multi-party control on access to the KRI.

All level 1 requirements apply in addition to the following:

~~(Req. 77)~~(Req. 80) The system shall be designed for multiple KRAs. Two or more KRAs shall be required for a requestor to obtain the target key.

3.1.2.4 Integrity

Same as Level 1.

3.1.2.5 Audit

Level 2 adds a real time alarm to ~~the~~ a security officer in the event that the audit trail becomes full in order to prevent audit data from being lost.

Includes all the requirements of Level 1 and the following:

~~(Req. 78)~~(Req. 81) The following actions shall be auditable:

(a) Execution of the tests of the underlying machine and the results of the tests; and

(b) Attempts to provide invalid inputs for administrative functions.

3.1.2.6 Identification and Authentication

Level 2 enhances assurance by requiring the use of a hardware token for user authentication. This provides an additional countermeasure to the threat of an attack on the authentication mechanism and the subsequent unauthorized access to KRI or critical functions. (Note: If ~~the~~ a crypto officer is invoking a KRA cryptographic module function, authentication may be effected directly to the module and is exempt from the following ~~newly added~~ requirement. In this case, the FIPS 140-1 level 3 module I&A requirements apply.)

All Level 1 requirements except that (Req. 55)~~(Req. 54)~~~~(Req. 52)~~ is replaced by the following:

~~(Req. 79)~~(Req. 82) The product shall support a hardware token-based authentication. The token shall meet FIPS 140-1 Level 2 requirements.

3.1.2.7 Access Control

Level 2 requires multi-party access controls for the release of KRI, and establishes roles and responsibilities for key recovery facility personnel as additional countermeasures to the threat of a single rogue Key Recovery Agent.

All Level 1 requirements apply as well as the following:

~~(Req. 80)~~(Req. 83) **The KRA Function shall be capable of requiring multi-party (at least 2) authorization in support of the release of target key information.**

Note that although the KRA must support multi-party authorization for the release of target key information, a product that may be configured to operate with single-party authorization would also be compliant.

The following requirements are intended to provide for strict role separation.

~~(Req. 81)~~(Req. 84) **The product shall define a set of security administrative roles that minimally includes a system administrator, a system operator, a crypto officer and an audit administrator.**

~~(Req. 82)~~(Req. 85) **~~The~~ An individual in the system administrator role shall perform the following functions:**

(a) the assignment/deletion of authorized users from system administrative roles,

(b) the association of security-relevant administrative commands with security administrative roles,

(c) the assignment/deletion of ~~subjects~~ subscribers whose keys are held, and

(d) the assignment/deletion of parties who may be provided the keys.

~~(Req. 83)~~(Req. 86) **The system operator shall change the system configuration and operate the system.**

~~(Req. 84)~~(Req. 87) **The crypto officer shall manage the cryptographic keys.**

~~(Req. 85)~~(Req. 88) **The audit administrator shall manage the audit log and audit profiles.**

~~(Req. 86)~~(Req. 89) **The product shall associate each security-relevant administrative function with ~~at least~~exactly one security administrative role.**

~~(Req. 87)~~(Req. 90) **The product shall enforce checks for valid input values for security-relevant administrative functions as described in the Administrative guidance.**

Note that the “Administrative guidance” document is a vendor-supplied document.

3.1.2.8 Authentication of Received Transactions

Same as Level 1.

3.1.2.9 Non Repudiation

Same as Level 1.

3.1.2.10 Protection of Trusted Security Functions

All Level 1 requirements apply as well as the following:

~~(Req. 88)~~(Req. 91) **The product shall provide the ~~authorized administrator~~system operator role with the capability to demonstrate the correct operation of the security-relevant functions provided by the underlying abstract machine.**

~~(Req. 89)~~(Req. 92) **The product shall preserve a secure state when the abstract machine tests fail.**

These two requirements ensure that the particular hardware system on which KRA software is operating is operating correctly. (Req. 91)~~(Req. 90)~~~~(Req. 88)~~ can be met by providing comprehensive integrity or diagnostic tests on the hardware. (Req. 92)~~(Req. 91)~~~~(Req. 89)~~ can be met by terminating the KRA operations in case of hardware integrity or diagnostic test failure.

3.2 Key Recovery Information Generation Function

3.2.1 Level 1 – Medium Assurance Key Recovery Information Generator

Note that these requirements are applicable to cryptographic end system products.

3.2.1.1 Cryptographic Functions

~~(Req. 90)~~(Req. 93) All cryptographic modules shall be FIPS 140-1, Level 1 compliant.

3.2.1.2 Cryptographic Algorithms

~~(Req. 91)~~(Req. 94) A KRI Generation Function submitted for evaluation shall be able to be configured to use only FIPS approved algorithms (where applicable).

See (Req. 25)~~(Req. 24)~~~~(Req. 21)~~ for additional clarifying details.

3.2.1.3 Confidentiality

This requirement is intended to minimize the vulnerability created by the key recovery mechanism. The key recovery mechanism should not be weaker and thus easier to attack than the original encryption mechanism.

~~(Req. 92)~~(Req. 95) Transmitted ~~target key information~~KRI must be protected via encryption. The strength of the algorithm used to protect the ~~target key~~KRI ~~information~~ shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.

3.2.1.4 Integrity

These requirements counter the threat of an outsider corrupting the KRI.

~~(Req. 93)~~(Req. 96) The KRI Generation Function shall generate an integrity value for the KRI.

~~(Req. 94)~~(Req. 97) The KRI Generation Function shall associate the KRI with the encrypted data.

~~(Req. 95)~~(Req. 98) The KRI Generation Function shall generate an integrity value for the association of the KRI to the data.

As an example, a key recovery scheme that includes a keyed message digest computed on the KRI using the data key meets all of the above three requirements. (Req. 96)~~(Req. 95)~~~~(Req. 93)~~ is met since the keyed message digest provides integrity. (Req. 97)~~(Req. 96)~~~~(Req. 94)~~ is met by the unambiguous placement of KRI and encrypted data as defined by the protocol (e.g., fixed location, pointer, tagged information, etc.). (Req. 98)~~(Req. 97)~~~~(Req. 95)~~ is met since the same key is used to calculate or verify the keyed message digest and to decrypt the data, which ensures the integrity of the association between the KRI and the encrypted data.

3.2.1.5 Identification and Authentication

~~(Req. 96)~~(Req. 99) All cryptographic modules shall implement role-based authentication.

~~(Req. 97)~~(Req. 100) One of the roles shall be the system administrator role.

3.2.1.6 Access Control

~~(Req. 98)~~(Req. 101) The KRI Generation Function shall allow only ~~the~~ a system administrator to configure this function.

~~(Req. 99)~~(Req. 102) At a minimum, the configurations shall include activation and deactivation of this function.

Note that a product in which KRI generation is always active need not meet the requirements of this section nor of Section 3.2.1.5.

3.2.2 Level 2 – High Assurance Key Recovery Information Generator

3.2.2.1 Cryptographic Functions

~~(Req. 100)~~(Req. 103) All cryptographic modules shall be FIPS 140-1, Level 2 compliant.

3.2.2.2 Cryptographic Algorithms

Same as Level 1.

3.2.2.3 Confidentiality

Same as Level 1.

3.2.2.4 Integrity

All of Level 1 requirements apply as well as the following:

~~(Req. 101)~~(Req. 104) The product shall generate KRI to allow the KRI Validation Function to verify that the KRI can be successfully used to recover the target key.

Note that an instance of a KRI Generation Function may not provide all of the data required for the KRI Validation Function.

3.2.2.5 Identification and Authentication

No requirements at this level.

3.2.2.6 Access Control

No requirements at this level.

3.3 Key Recovery Information Delivery Function

No Security requirements.

3.4 Key Recovery Information Validation Function

Note that a KRS composed from Level 1 products need not include a KRI Validation Function.

3.4.1 Level 1 – Medium Assurance Key Recovery Information Validation Function

3.4.1.1 Cryptographic Functions

~~(Req. 102)~~(Req. 105) All cryptographic modules shall be FIPS 140-1, Level 1 compliant.

3.4.1.2 Cryptographic Algorithms

~~(Req. 103)~~(Req. 106) A KRI Validation Function which is submitted for evaluation shall be able to be configured to use only FIPS approved algorithms (where applicable).

3.4.1.3 Integrity

The purpose of the integrity requirements is to ensure that the KRI can be used to successfully decrypt the communication when the receiver can successfully decrypt the communication. Level 1 requirements counter the threat of an outsider corrupting the KRI. Level 2 requirements counter the threat of the sender corrupting the KRI.

__________ END OF INITIAL REVIEW OF THE TEXT BY THE TAC _________

~~(Req. 104)~~(Req. 107) The KRI Validation Function shall be configurable.

In order to facilitate interoperability due to differences in key recovery schemes, levels of functionality, and/or configuration (e.g., whether or not key recovery is enabled), this function needs be configurable. If ~~integrity~~ KRI validation is ~~enabled~~ enabled (i.e., turned on), it may prevent interoperation between two cryptographic end systems. [move requirement #104 and explanatory text to section 2.3]

~~(Req. 105)~~(Req. 108) Prior to decrypting the data, the KRI Validation Function (if enabled) shall verify the integrity value for ~~that the KRI acquired was that intended by~~ the KRI ~~Generation Function~~.

~~(Req. 106)~~(Req. 109) Prior to decrypting the data, the KRI validation Function (if enabled) shall verify ~~that~~ the association of the KRI with the encrypted data ~~was that intended by the KRI Generation Function~~.

~~(Req. 107)~~(Req. 110) Prior to decrypting the data, the KRI Validation Function (if enabled) shall verify the integrity value for~~of~~ the association of the KRI to the encrypted data.

See Section 3.2.1.4 "Key Recovery Information Generation Function – Integrity" for an example of how the above integrity requirements can be satisfied.

3.4.2 Level 2 – High Assurance Key Recovery Information Validator

3.4.2.1 Cryptographic Functions

~~(Req. 108)~~(Req. 111) All cryptographic modules shall be FIPS 140-1, Level 2 compliant.

3.4.2.2 Cryptographic Algorithms

Same as Level 1.

3.4.2.3 Integrity

~~The product shall meet at least one of the following (i.e., is required to meet only one of the following, but may meet more than one) integrity requirements:~~

~~(Req. 109)~~(Req. 112) When interoperating with another product implementing the same key recovery scheme, the product shall meet at least one of the following requirements. Otherwise the product needs to meet only the Level 1 integrity requirements.

1. **The KRI Validation Function shall ensure that the KRI received is accurate, i.e., the information can be used to perform key recovery successfully.**

~~(Req. 110)~~2. A KRI Generation Function in the receiving cryptographic end system shall generate accurate key recovery information for received encrypted data.

~~(Req. 111)~~3. The receiving cryptographic end system shall not be able to obtain the correct data decryption key if the received key recovery information is not accurate.

3.5 Key Recovery Requestor Function

~~The security requirements for the Key Recovery Requestor Functions have been defined to allow a variety of product architectures. These include using a monolithic product on which no other software/firmware can be loaded, using a monolithic product on which other software/firmware can be loaded, or using a layered product that has a distinct operating system, application, and cryptographic module.~~

~~The requirements for the Key Recovery Requestor Functions have been defined so that all of these architectures can be evaluated. This is especially true of the requirements in the following areas: Audit, Identification and Authentication, Access Control, and Protection of Trusted Security Functions.~~

~~Furthermore, the product architecture may imply that some of the requirements do not apply, e.g., if the threat that a requirement is intended to mitigate does not arise in a particular implementation model. For example, if the product is a monolithic product on which no other software/firmware can be loaded, the domain separation, trusted path, and reference validation mechanism requirements do not apply since the untrusted software threat does not exist.~~
For this function, a third, lower level of security requirements is defined. The primary motivation for this additional level is self-recovery.

1.1.1 Level 0- Low Security

3.5.1.1 Cryptographic Functions

(Req. 113) All cryptographic modules shall be compliant with FIPS 140-1, Level 1 or higher.

3.5.1.2 Cryptographic Algorithms

(Req. 114) **A Key Recovery Requestor function submitted for evaluation shall be able to be configured to use only FIPS approved algorithms (where applicable).**

If a cryptographic function can be effected using a FIPS approved algorithm, it must be possible to configure the requestor to make use of this algorithm. However, if a key recovery scheme requires a cryptographic function not supported by any FIPS approved algorithms, there is no requirement to make use of such algorithm, e.g., use of RSA[10]for key encapsulation.

3.5.1.3 Confidentiality

There are no confidentiality requirements imposed at this level.

3.5.1.4 Integrity

(Req. 115) **The product shall apply data origin authentication to all requests. The strength of the algorithm used for authentication shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.**

(Req. 116) **The product shall apply integrity services to all requests. The strength of the algorithm used for integrity shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.**

3.5.1.5 Audit

There are no audit requirements imposed at this level.

3.5.1.6 Identification and Authentication

There are no I&A requirements imposed at this level.

3.5.1.7 Access Control

There are no access control requirements imposed at this level.

[10] ANSI X9.31, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)

3.5.1.8 Authentication of Received Transactions

There are no authentication of received transactions requirements imposed at this level.

3.5.1.9 Non-Repudiation

(Req. 117) The product shall provide time stamps for use in transactions with the KRA Function.

(Req. 118) The product shall generate evidence of origin for key recovery requests.

3.5.1.10 Protection of Trusted Security Functions

There are no protection of trusted security functions requirements imposed at this level.

3.5.1 Level 1 – Medium Assurance

[now 3.5.2, make each subsection of this relative to 3.5.1]

3.5.1.1 Cryptographic Functions

~~(Req. 112)~~(Req. 119) All cryptographic modules shall be compliant with FIPS 140-1, Level 2 or higher.

3.5.1.2 Cryptographic Algorithms

~~(Req. 113)~~(Req. 120) A Key Recovery Requestor function submitted for evaluation shall be ~~be~~ able to be configured to use only FIPS approved algorithms (where applicable).

If a cryptographic function can be effected using a FIPS approved algorithm, it must be possible to configure the requestor to make use of this algorithm. However, if a key recovery scheme requires a cryptographic function not supported by any FIPS approved algorithms, there is no requirement to make use of such algorithm, e.g., use of RSA[11] for key encapsulation.

[11] ANSI X9.31, Digital Signatures Using Reversible Public Key Cryptography for the Financial Services Industry (rDSA)

3.5.1.3 Confidentiality

~~(Req. 114)~~(Req. 121) The requestor shall protect both received and~~/or~~ stored ~~KRI~~ TKI against disclosure to unauthorized individuals.

Note: Storing the ~~data~~ TKI in encrypted form or~~and~~ implementing access controls ~~is~~ are ~~one~~ two examples of ways to meet this requirement.

~~(Req. 115)~~(Req. 122) The requestor shall protect the key recovery request (especially the identities of subjects and time periods, if applicable) transmitted against disclosure to parties other than the KRA.

Note: Encryption of the request is one way to meet this requirement.

(Req. 123) If a requestor is required, by policy, to notify other parties when key recovery requests are performed, such notifications shall be protected against unauthorized disclosure.

Note: Encryption of the notification is one way to meet this requirement.

~~(Req. 116)~~(Req. 124) The product shall apply confidentiality services to all requests and notifications. The strength of the algorithm used for confidentiality shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.

3.5.1.4 Integrity

~~(Req. 117)~~(Req. 125) The product shall apply data origin authentication to all requests. The strength of the algorithm used for authentication shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.

~~(Req. 118)~~(Req. 126) The product shall apply integrity services to all requests. The strength of the algorithm used for integrity shall be greater than or equal to the strength of the encryption and key management algorithms employed for data encryption or for generation of the keys being recovered.

3.5.1.5 Audit

These requirements are used to create a log of information to allow oversight by a security officer to detect unauthorized operations by a Key Recovery Requestor. The recording of events defined as "auditable" may be enabled under configuration control.

~~(Req. 119)~~(Req. 127) The Key Recovery Requestor (KRR) shall cease operation if it is unable to effect audit operations.

~~(Req. 120)~~(Req. 128) The product shall generate an alarm to ~~the~~ an authorized administrator if the size of the audit data in the audit trail exceeds a pre-defined limit.

~~(Req. 121)~~(Req. 129) The product shall provide ~~the~~ an authorized administrator with the ability to manage the audit trail at any time during the operation of the product.

~~(Req. 122)~~(Req. 130) Keys shall not be included in audit trails.

~~(Req. 123)~~(Req. 131) The following actions shall be auditable:

(a) Any specific operation performed to process audit data stored in the audit trail; (Note: This include backup and deletion of audit trail)
(b) Any attempt to read, modify or destroy the audit trail;
(c) All requests to use authentication data management mechanisms;
(d) All modifications to the audit configuration that occur while the audit collection functions are operating;
(e) All requests to access user authentication data;
(f) Any use of an authentication mechanism. (e.g. login);
(g) All attempts to use the user identification mechanism, including the user identity provided;
(h) Use of a security-relevant administrative function;
(i) Explicit requests to assume ~~the~~ a security administrative role;
(j) The allocation of a function to a security administrative role;
(k) The addition or deletion of a user to/from a security administrative role;
(l) The association of a security-relevant administrative function with a specific security administrative role.

(m) The invocation of the non-repudiation service. The audit event shall include the identification of the information, the destination, and a copy of the evidence provided. The event shall exclude all private and secret keys in encrypted or unencrypted form.

(n) All attempted uses of the trusted path functions; and

(o) Identification of the initiator and target of the trusted path.

~~(Req. 124)~~(Req. 132) It shall not be possible to disable ~~T~~the ~~recording~~ auditing of an event defined as "always audited." ~~shall not be disable-able.~~

~~(Req. 125)~~(Req. 133) The following events shall always be audited:

(a) Requests, ~~responses~~notifications, and other transactions generated by the product~~, including key recovery responses~~;

(b) R~~equests, r~~esponses~~,~~ and other transactions received by the product, including key recovery ~~requests~~responses; and

(c) Start-up and shutdown of the audit functions.

~~(Req. 126)~~(Req. 134) The product shall record at least the following information within each audit record:

(a) Date and time of the event, type of event, subject (user) identity, and success or failure of the event; and

(b) Other audit event type information as follows:

(1) For changes to the configuration file event, changes shall also be recorded in the audit record.

(2) When attempting a function using ~~the~~ a security administrative role, the function attempted, the role and all applicable inputs shall be recorded in the audit record.

(3) When allocating a function to a security administrative role, the role and the function shall be included in the audit record.

(4) When adding or deleting users to/from ~~the~~ a security administrative role, the role, user identity and the addition/deletion action shall be included in the audit record.

(5) For all ~~KRA~~ transactions, the entire transaction (excluding keys and TKI) shall be included in the audit record as sent or received.

~~(Req. 127)~~(Req. 135) The product shall be able to generate a human understandable presentation of any audit data stored in the permanent audit trail.

~~(Req. 128)~~(Req. 136) The audit trail shall not store the old or new authentication information (e.g., passwords)

~~(Req. 129)~~(Req. 137) The product shall be able to associate each auditable event with the identity of the user that caused the event.

~~(Req. 130)~~(Req. 138) The product shall provide ~~the~~ an authorized administrator with the ability to empty the audit trail.

~~(Req. 131)~~(Req. 139) The product shall be able to include or exclude auditable events from the set of audited events based on the following attributes: u~~U~~ser identity~~,~~ and/or e~~E~~vent t~~T~~ype.

~~(Req. 132)~~(Req. 140) The product shall restrict access to the audit trail to ~~the~~ authorized administrators.

3.5.1.6 Identification and Authentication

The requirements in this section are for the identification and authentication of ~~the various~~ requestor personnel. This facilitates individual accountability via audit functions and access controls. Requirements are levied on the strength of the authentication mechanism against attacks by rogue KRR personnel.

These requirements do not apply to electronic transactions (requests and responses). The electronic transactions may be identified and authenticated (if the scheme permits) using the access control policy.

Note: If ~~the~~ a crypto officer is invoking a KRR cryptographic module function, authentication may be effected directly to the module and is exempt from ~~the~~ all of the ~~following newly added~~ requirements of this section. In this case, the FIPS 140-1 level 2 module I&A requirements apply.

~~(Req. 133)~~(Req. 141) The product shall provide functions for initializing and modifying ~~user~~ KRR personnel authentication data.

~~(Req. 134)~~(Req. 142) The product shall restrict the use of initialization and modification of the ~~user~~ KRR personnel authentication data ~~to~~ to a security administrators.

~~(Req. 135)~~(Req. 143) The product shall allow authorized ~~users~~ KRR personnel to ~~use these functions to~~ modify their own authentication data.

~~(Req. 136)~~(Req. 144) **The product shall protect authentication data that is stored in the product from unauthorized observation, modification, and destruction.**

~~(Req. 137)~~(Req. 145) **The product shall protect authentication information from unauthorized reuse, including replay.**

Note: This requirement and the previous requirement provide a capability for secure remote login.

~~(Req. 138)~~(Req. 146) **The product shall be able to terminate the ~~the user~~ session establishment process after at most five unsuccessful authentication attempts~~.~~.**

~~(Req. 139)~~(Req. 147) **After the termination of ~~a~~ the ~~user~~ session establishment process, the product shall be able to disable the ~~user~~ KRR personnel account until the account is enabled by an authorized administrator (i.e., a security administrator).**

~~(Req. 140)~~(Req. 148) **The product shall authenticate ~~every user's~~ the claimed identity of an individual prior to performing any functions on ~~the user's~~ behalf of that individual.**

~~(Req. 141)~~(Req. 149) **The product shall require a user authentication technology that protects authentication information capture (this requirement is met by a trusted path or the use of a one time password). The strength of the mechanism ~~in terms of space shall meet the requirement of 1 in 1,000,000~~ shall nominally reduce the likelihood of false authentication to less than 1/1,000,000.**

Techniques that meet this requirement are defined in FIPS PUB 112 based passwords entered via a trusted path, RFC 1938 (One Time Password), hardware tokens connected via trusted channels/paths, and biometric tokens connected via trusted channels/paths.

~~(Req. 142)~~(Req. 150) **If the product makes use of a "trusted path" mechanism to meet the preceding I&A requirement, that trusted path between itself and ~~local human users~~KRR personnel shall be logically distinct from other communication paths and shall provide an assured identification of its endpoints. ~~The local human user~~KRR personnel shall have the ability to initiate communication via this trusted path.**

3.5.1.7 Access Control

~~These requirements provide countermeasures against an entity masquerading as an authorized requestor. The requirements in this section address the security of electronic communication between the KRA and Key Recovery Requestor Functions. If these interactions are not electronic, then physical and procedural means may be used to secure the transactions. These procedural and physical measures are beyond the scope the standard.~~

~~(Req. 143)~~(Req. 151) The product shall verify the association of the response to an outstanding request.

~~(Req. 144)~~(Req. 152) The product shall ~~ensure~~ provide an ability to destroy ~~that the KRI~~ TKI and target keys, ~~is destroyed (~~e.g., by zeroizing.~~)~~ ~~when it is no longer required, when it is no longer valid (e.g., time expiry), when the KRA requires its deletion, or when the legal authority to it expires, whichever occurs first.~~

Destruction of this data may be performed when it is no longer required, no longer valid (e.g., time expiry), when the KRA requires its deletion, or when the authority to possess it expires.

~~(Req. 145)~~(Req. 153) The product shall ensure that security features are always invoked and cannot be bypassed.

~~(Req. 146)~~(Req. 154) The product shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

~~(Req. 147)~~(Req. 155) The product shall enforce separation between the security domains of subjects in the system.

~~(Req. 148)~~(Req. 156) The product shall restrict the ability to perform security-relevant administrative functions to a security administrative role that has a specific set of authorized functions and responsibilities.

Note: The term “security administrative role” refers to generic trusted administrative roles. The system administrator role is one, but not the only one, of these security administrative roles. Additional security administrative roles are defined in Requirement (Req. 84)~~(Req. 83)(Req. 81)~~.

In order to meet the preceding requirements, the product must distinguish security-relevant administrative functions from other administrative functions. The set of security-relevant administrative functions must include all functions necessary to install, configure, and manage the product; minimally, this set must include:

- the assignment/deletion of authorized users from security administrative roles,
- the association of security-relevant administrative commands with security administrative roles,
- the assignment/deletion of ~~subjects~~ authorized requestors ~~whose keys are held~~,
- ~~the assignment/deletion of parties who may be provided the keys,~~
- product cryptographic key management,
- actions on the audit log, audit profile management, and
- changes to the system configuration.

~~(Req. 149)~~(Req. 157) The product shall be capable of distinguishing the set of ~~users~~ KRR personnel authorized for administrative functions from ~~the set of~~ all other ~~users~~personnel.

~~(Req. 150)~~(Req. 158) The product shall allow only specifically authorized ~~users~~ KRR personnel to assume ~~the~~ a security administrative role.

~~(Req. 151)~~(Req. 159) The product shall require an explicit request to be made in order for an authorized ~~users~~ KRR personnel to assume ~~the~~ a security administrative role.

3.5.1.8 Authentication of Received Transactions

~~(Req. 152)The product shall verify the source of received transactions.~~

~~(Req. 153)The product shall verify the integrity of received transactions.~~
Same requirements as at Level 0.

3.5.1.9 Non-Repudiation

Same as Level 0, except <reference next requirement> replaces <reference ??>.

~~(Req. 154)~~(Req. 160) The product shall provide trusted time stamps for use in transactions with the KRA Function.

~~(Req. 155)The product shall verify evidence of origin for key recovery responses.~~

~~(Req. 156)The product shall generate evidence of origin for key recovery requests.~~

3.5.1.10 Protection of Trusted Security Functions

~~(Req. 157)~~(Req. 161) Before establishing a session with an individual~~ user~~, the product shall display an advisory warning message regarding unauthorized use of the product.

~~(Req. 158)~~(Req. 162) The default advisory warning message displayed by the product shall be as follows: "This system shall be used only by authorized personnel and only for authorized key recovery purposes. Violation may result in criminal prosecution and civil penalties".

~~(Req. 159)~~(Req. 163) The product shall restrict the capability to modify the warning message to ~~the~~ an authorized security administrative~~or~~ role.

~~(Req. 160)~~(Req. 164) Upon successful session establishment, the product shall display the date, time, method, and location of the last successful session establishment ~~to~~ the ~~user~~individual establishing the session.

~~(Req. 161)~~(Req. 165) Upon successful session establishment, if there have been any unsuccessful session establishment attempts since the last successful session establishment, the product shall display the date, time, method, and location of the most recent unsuccessful attempt to session establishment as well as the number of unsuccessful attempts since the last successful session establishment.

~~(Req. 162)~~(Req. 166) The data specified above shall not be removed without ~~user~~ intervention by the individual establishing the session.

3.5.2 Level 2 – High Assurance

3.5.2.1 Cryptographic Functions

~~(Req. 163)~~(Req. 167) All cryptographic modules shall be compliant with FIPS 140-1, Level 3 or higher.

3.5.2.2 Cryptographic Algorithms

Same as Level 1.

3.5.2.3 Confidentiality

Same as ~~level~~ Level 1.

3.5.2.4 Integrity

Same as ~~level~~ Level 1.

3.5.2.5 Audit

Includes all the requirements of Level 1 and the following:

~~(Req. 164)~~(Req. 168) The following actions shall be auditable:

- **(a) Execution of the tests of the underlying machine and the results of the tests;**
- **(b) Attempts to provide invalid inputs for administrative functions**.

3.5.2.6 Identification and Authentication

Level 2 enhances assurance by requiring the use of a hardware token for user authentication. This provides an additional countermeasure to the threat of an attack on the authentication mechanism and the subsequent unauthorized access to KRI or critical functions. (Note: If ~~the~~ a crypto officer is invoking a KRA cryptographic module function, authentication may be effected directly to the module and is exempt from the following ~~newly added~~ requirement. In this case, the FIPS 140-1 level 3 module I&A requirements apply.)

All Level 1 requirements except that (Req. 149)~~(Req. 151)~~~~(Req. 141)~~ is replaced by the following: ~~THE NUMBER 52 NEEDS TO BE CHANGED TO A NUMBER IN KRR.~~

~~(Req. 165)~~(Req. 169) The product shall support ~~a~~ hardware token-based authentication. The token shall meet FIPS 140-1 Level 2 requirements.

3.5.2.7 Access Control

All Level 1 requirements apply as well as the following:

~~(Req. 166)~~(Req. 170) Two or more ~~users~~ individuals shall be required to request the ~~recovery information~~TKI from the KRA Function.

~~(Req. 167)~~(Req. 171) The product shall define a set of security administrative roles that minimally includes a system administrator, a system operator, a crypto officer, and an audit administrator.

~~(Req. 168)~~(Req. 172) An individual in ~~T~~the system administrator role shall perform the following functions:

(a) the assignment/deletion of KRR personnel accounts,

~~(a)~~(b) the assignment/deletion of authorized ~~users~~ KRR personnel to/from ~~system~~ security administrative roles, and

~~(b)~~(c) the association of security-relevant administrative commands with security administrative roles.~~,~~

~~(c)the assignment/deletion of subjects whose keys are held, and~~

~~(d)~~**~~the assignment/deletion of parties who may be provided the keys~~.**

~~(Req. 169)~~(Req. 173) The system operator shall be able to change the system configuration, execute abstract machine tests, change the advisory warning message, and operate the system.

~~(Req. 170)~~(Req. 174) The crypto officer shall manage the cryptographic keys.

~~(Req. 171)~~(Req. 175) The audit administrator shall manage the audit log and audit profiles.

~~(Req. 172)~~(Req. 176) The product shall associate each security-relevant administrative function with ~~at least~~ exactly one security administrative role.

~~(Req. 173)~~(Req. 177) The product shall enforce checks for valid input values for security-relevant administrative functions as described in the Administrative guidance.

Note that the “Administrative guidance” document is a vendor-supplied document.

3.5.2.8 Authentication of Received Transactions

Same as Level 1.

3.5.2.9 Non Repudiation

Same as Level 1~~ requirements~~.~~:~~

3.5.2.10 Protection of Trusted Security Functions

All Level 1 requirements apply as well as the following:

~~(Req. 174)~~(Req. 178) The product shall provide the system operator~~authorized administrator~~ role with the capability to demonstrate the correct operation of the security-relevant functions provided by the underlying abstract machine.

~~(Req. 175)~~(Req. 179) The product shall preserve a secure state when abstract machine tests fail.

These two requirements ensure that the particular hardware system on which ~~KRA~~ KRR software is operating is operating correctly. (Req. 91)~~(Req. 90)(Req. 88)~~ can be met by providing comprehensive integrity or diagnostic tests on the hardware. (Req. 92)~~(Req. 91)(Req. 89)~~ can be met by terminating the ~~KRA~~ KRR operations in case of hardware integrity or diagnostic test failure. [Elaine: Numbers 88 and 89 in this text need to refer to KRR requirements.]

~~THE FOLLOWING ARE NOT REQUIREMENTS, BUT INFORMATIVE TEXT TO BE USED SOMEWHERE OR TO BE DELETED.~~

~~KRA Availability~~

~~These suggestions are intended to provide the capability for a KRA to recover in the event of a system failure or compromise. They act as a counter to the threat of the unauthorized destruction of the KRI or capabilities at the KRA facility.~~

~~The KRA facility should be required to have the capability to securely replicate any KRI stored in order to support continued on-line access in case of a facility failure.~~

~~The KRA facility should have a secure backup of the KRI stored in order to rebuild the key recovery database in case of KRA system failure.~~

~~Ancillary Products~~

~~Registration Agent~~

~~Registration Agents maintain information on key recovery products and corresponding key recovery protocol (schemes). The registration agent should be able to ensure the accuracy and maintain the integrity of the product information.~~

~~Integrity/Authenticity~~

~~These features counter the threat of an adversary spoofing as the registration agent and of unauthorized access to the information and critical functions at the registration agent.~~

~~The Registration Agent should verify authentication and integrity services for the received product information.~~

~~The Registration Agent should apply authentication and integrity services to the product information it transmits.~~

~~The Registration Agent should ensure that the product information it maintains is not modified by unauthorized parties.~~

~~Licensing Agent~~

~~Licensing Agents perform compliance audits of the KRAs to ensure that the KRAs operate in accordance with the KRA's stated policy.~~

~~Authentic Public Key Source (APKS AKA Public Key Infrastructure (PKI))~~

~~Standards~~

~~The APKS should carry out transactions in accordance with the Minimum Interoperability Specifications for PKI Products (MISPC)~~

~~Security/Certificate Policy:~~

~~The security of PKI and the degree to which the binding between an entity (subject or subscriber) and public key can be trusted, is determined by the Certificate Policy. Certificate Policy is defined and described in Certificate Policy Framework. Using this Framework, NIST has developed Baseline Security Requirements. NIST plans to enhance these for up to three more strictly superior policies. Thus, in order to define the security requirements for the APKS, we only need to select the proper certificate policy. Please note the certificate policy security requirements are quite comprehensive. For details, see IETF PKIX Part IV: Certificate Policy Framework.~~

~~For Level 1, the APKS should meet the medium Certificate Policy. For Level 2, the APKS should meet the high Certificate Policy.~~

4 Assurance Requirements

The assurance in a KRS compliant product can be achieved using the Common Criteria Evaluation Assurance Levels (EAL). The Common Criteria (CC) defines seven hierarchical assurance levels EAL1 through EAL7. The Common Criteria assurance levels appear ~~may be overkill~~ excessive for the KRS compliance validation program. Thus, this section contains a tailored list of assurance requirements. These requirements are derived from the Common Criteria Part 3 (Assurance Requirements). Specifying assurance requirements in the common criteria language ~~will help~~aids in converting the FIPS into a Common Criteria Protection Profile and in validating KRS compliant products under the Common Criteria (CC) Evaluation Methodology. Section <reference to 4.8> explains why some portions of Common Criteria assurance requirements are not recommended.

~~For the sake of clarity, it should be noted that the CC structure for assurance requirements is hierarchical as follows. At the highest level, the requirements are categorized into classes. The classes are further decomposed into families. The families are decomposed into products. Each component has three sets of elements. The first set of elements is the list of developer (vendor) requirements which must be satisfied for the component. The second set of elements is a list of contents and presentation requirements for the assurance evidence for that element. The third and last set of elements is what an independent evaluator should do to assess the contents and presentation items which are provided.~~

~~A later section of this report also explains why the remaining Common Criteria assurance requirements are not recommended.~~

Three Assurance Levels (ALs) are defined for this standard. These levels are ~~somewhat~~ related to the Common Criteria EALs ~~assurance levels~~, but are not derived from the Common Criteria EALs ~~assurance levels~~. ~~The~~ Table 1 ~~assurance levels~~ contains ~~for~~ the classes, families, and ~~products~~ components for the three ALs~~of key recovery products are listed in Table 1~~. Subsequent sections provide further detail.

~~(Req. 176)~~(Req. 180) The KRA ~~and Key Recovery Requestor~~ Function~~s~~ shall be required to meet the assurance requirements for AL B and AL C for Security Levels 1 and 2, respectively, as defined in Tables 1 and 2.

(Req. 181) The KRR Function shall be required to meet the assurance requirements for AL A, AL B, and AL C for Security Levels 0, 1, and 2, respectively, as defined in Tables 1 and 2.

~~(Req. 177)~~(Req. 182) The KRI Generation and Validation Functions shall be required to meet the assurance requirements for AL A and AL B for Security Levels 1 and 2, respectively, as defined in Tables 1 and 2.

Table 2 provides a summary of assurance level requirements for the various KRS functions.

It should be noted that the assurance requirements are applied to test ~~the~~ product functionality and security features.

Assurance Concept

The assurance concepts and notations in this standard are based on the Common Criteria. The assurance concept consists of a hierarchical refinement of the requirements. At the top-level, the assurance requirements are broken down into classes. The classes include, configuration management, delivery and operation, development, guidance documents, life-cycle support, testing, and vulnerability analysis. Each class is broken down into families. For example, the development class contains families such as functional specification, high-level design, low-level design, implementation representation, etc. Each family consists of one or more products. Each component contains three sets of elements. ~~The first set is the product developer actions. The second set is the requirements for content and presentation of information. The third and final set contains the evaluator actions.~~

~~**Assurance Notations**~~

~~The notation used for assurance requirements is based on the Common Criteria. Each class is defined as three characters; the first character is always "A" for assurance; the remaining two characters are meaningful for the class; e.g., CM for configuration management, DV for development, TE for testing, etc. The three letter assurance class is followed by an underscore "_" and a three letter meaningful name for the family, e.g., FSP for functional specification. The family is followed by "." and a component number. The component number is followed by a "." and a two character element indicator. Each component contains three sets of numbered elements. The first character of the element indicator is a sequential number within the set. The second character indicates the set : "D" for a developer action, "C" for content and presentation, and "E" for an evaluator action.~~

~~In Table 1 below, the numbers in the last three columns identify the products of each family that must be satisfied in order to provide the appropriate assurance level. For example, for assurance family **ADV_FSP** (see Section 4.3.1), assurance level A specifies that component 1 applies. Component 1 is identified as **ADV_FSP.1**, and the requirements are listed in Section 4.2.1.1, ADV_FSP.1 Functional Specification and Security Policy. For assurance levels B and C, component 2 applies. Component 2 is identified as **ADV_FSP.2**, and the requirements are listed in Section 4.3.1.2, ADV_FSP.2 Informal Security Policy Model.~~

Table 1: KRS Assurance Levels

Assurance Class	Assurance Family	AL A	AL B	AL C
Configuration Management	~~ACM_CAP~~ CM Capabilities	▨	1	1
	~~ACM_SCP~~ CM Scope	▨	▨	2
Delivery and Operation	~~ADO_DEL~~ Delivery	▨	1	2
	~~ADO_IGS~~ Installation, Generation and Start-up	1	1	1
Development	~~ADV_FSP~~ Functional Specification	1	2	2
	~~ADV_HLD~~ High-Level Design	1	2	2
	~~ADV_IMP~~ Implementation Representation	▨	▨	1
	~~ADV_LLD~~ Low-Level Design	▨	▨	1
	~~ADV_RCR~~ Representation Correspondence	▨	▨	1
Guidance Documents	~~AGD_ADM~~ Administrator Guidance	1	1	1
	~~AGD_USR~~ User Guidance	1	1	1
Life Cycle Support	~~ALC_FLR~~ Flaw Remediation	1	2	2
Tests	~~ATE_COV~~ Coverage	1	1	1
	~~ATE_DPT~~ Depth	1	1	1
	~~ATE_FUN~~ Functional Tests	1	1	1
	~~ATE_IND~~ Independent Testing	2	2	3
Vulnerability Assessment	~~AVA_VLA~~ Vulnerability Analysis	▨	1	1

Table 2: Assurance Levels for KRS Functions

KRS Function	Security Level 0	Security Level 1	Security Level	2
KRA	N/A	AL B	AL C	
Key Recovery Requestor	AL A	AL B	AL C	
KRI Generation	N/A	AL A	AL B	
KRI Delivery	N/A	AL A	AL B	
KRI Validation	N/A	AL A	AL B	

4.1 Configuration Management

Configuration management (CM) is an aspect of establishing that the functional requirements and specifications are realized in the implementation. CM meets these objectives by requiring discipline and control in the processes of refinement and modification of the product. CM systems are put in place to ensure the integrity of the configuration items that they control, by providing a method of tracking these configuration items, and by ensuring that only authorized users are capable of changing the items.

4.1.1 Configuration Management ACM_CAP – CM Capabilities

Objectives
The capabilities of the CM system address the likelihood that accidental or unauthorized modifications of the configuration items will occur. The CM system should ensure the integrity of the product from the early design stages through all subsequent maintenance efforts. The objectives of this assurance requirement include the following:

1. ensuring that the product is correct and complete before it is sent to the consumer; and
2. ensuring that no configuration items are missed during evaluation.

Clear identification of the product is required to determine those items under evaluation that are subject to the criteria requirements.

Application notes
There is a requirement that a configuration list be provided. The configuration list contains all configuration items ~~which are~~ maintained by the CM system.

4.1.1.1 ACM_CAP.1 Minimal Support

Developer action elements:

~~(Req. 178)~~(Req. 183) ACM_CAP.1.1D: **The developer shall use a CM system.**

~~(Req. 179)~~(Req. 184) ACM_CAP.1.2D: **The developer shall provide CM documentation.**

Content and presentation of evidence elements:

~~(Req. 180)~~(Req. 185) ACM_CAP.1.1C: **The CM documentation shall include a configuration list.**

~~(Req. 181)~~(Req. 186) ACM_CAP.1.2C: **The configuration list shall describe the configuration items that comprise the product.**

~~(Req. 182)~~(Req. 187) ACM_CAP.1.3C: **The CM documentation shall describe the method used to uniquely identify the product configuration items.**

Evaluator action elements:

~~(Req. 183)~~(Req. 188) ACM_CAP.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.1.2 Configuration Management ACM_SCP - CM Scope

Objectives

The objective is to ensure that all necessary configuration items are tracked by the CM system. This helps to ensure that the integrity of these configuration items is protected through the capabilities of the CM system. The objectives of this assurance requirement include the following:

1. ensuring that the implementation representation (i.e., code) is tracked; and
2. ensuring that all necessary documentation, including problem reports, are tracked during development and operation.

A CM system can control changes only to those items that have been placed under CM. The implementation representation, design, tests, user and administrator documentation, security flaws, and CM documentation should be placed under CM. The ability to track security flaws under CM ensures that security flaw reports are not lost or forgotten, and allows a developer to track security flaws to their resolution.

Application notes

There is a requirement that the implementation representation be tracked by the CM system. The implementation representation refers to all hardware, software, and firmware that comprise the physical product. In the case of a software-only product, the implementation representation may consist solely of source and object code, but in other cases, the implementation representation may refer to a combination of software, hardware, and firmware. There is a requirement that security flaws be tracked by the CM system. This requires that information regarding previous security flaws and their resolution be maintained, as well as details regarding current security flaws.

4.1.2.1 ACM_SCP.2 Problem Tracking CM Coverage

Developer action elements:

~~**(Req. 184)**~~**(Req. 189)** ACM_SCP.2.1D: **The developer shall provide CM documentation.**

Content and presentation of evidence elements:

~~**(Req. 185)**~~**(Req. 190)** ACM_SCP.2.1C: **As a minimum, the following shall be tracked by the CM system: the implementation representation, design documentation, test documentation, user documentation, administrator documentation, CM documentation, and security flaws.**

~~**(Req. 186)**~~**(Req. 191)** ACM_SCP.2.2C: **The CM documentation shall describe how configuration items are tracked by the CM system.**

Evaluator action elements:

~~**(Req. 187)**~~**(Req. 192)** ACM_SCP.2.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.2 Delivery and Operation

4.2.1 Delivery and Operation ADO_DEL – Delivery

Objectives

The requirements for delivery call for system control and distribution facilities and procedures that provide assurance that the recipient receives the product that the sender intended to send, without any modifications. For a valid delivery, what is received must correspond precisely to the master copy, thus avoiding any tampering with the actual version, or substitution of a false version.

Application notes
This assurance requirement should be applied to sensitive products whose modification can compromise security.

4.2.1.1 ADO_DEL.1 Delivery Procedures

Developer action elements:

~~(Req. 188)~~(Req. 193) ADO_DEL.1.1D: **The developer shall provide documentation about the procedures for the delivery of the product or parts of the product to the user.**

~~(Req. 189)~~(Req. 194) ADO_DEL.1.2D: **The developer shall use the delivery procedures. [NOTE: IS THIS TESTABLE**?]

Content and presentation of evidence elements:

~~(Req. 190)~~(Req. 195) ADO_DEL.1.1C: **The delivery documentation shall describe the procedures to be employed when distributing versions of the product to a user's site.**

Evaluator action elements:

~~(Req. 191)~~(Req. 196) ADO_DEL.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.2.1.2 ADO_DEL.2 Detection of Modification

Developer action elements:

~~(Req. 192)~~(Req. 197) ADO_DEL.2.1D: **The developer shall provide documentation about the procedures for the delivery of the product or parts of the product to the user.**

~~(Req. 193)~~(Req. 198) ADO_DEL.2.2D: **The developer shall use the delivery procedures.** [NOTE: IS THIS TESTABLE?]

Content and presentation of evidence elements:

~~(Req. 194)~~(Req. 199) ADO_DEL.2.1C: **The delivery documentation shall describe the procedures to be employed when distributing versions of the product to a user's site.**

~~(Req. 195)~~(Req. 200) ADO_DEL.2.2C: **The delivery documentation shall state how the procedures are to be employed to detect modifications.**

~~(Req. 196)~~(Req. 201) ADO_DEL.2.3C: **The delivery documentation shall describe how the various procedures and technical measures provide for the detection of modifications, or any discrepancy between the developer's master copy and the version received at the user site.**

~~(Req. 197)~~(Req. 202) ADO_DEL.2.4C: **The delivery documentation shall describe how the various procedures allow the detection of attempted masquerading even in cases in which the developer has sent nothing to the user's site.**

Evaluator action elements:

~~(Req. 198)~~(Req. 203) ADO_DEL.2.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.2.2 Delivery and Operation ADO_IGS - Installation, Generation, and Start-up

Objectives

Installation, generation, and start-up procedures are useful for ensuring that the product has been installed, generated, and started in a secure manner as intended by the developer.

Application notes

The generation requirements are applicable only to the products that provide the ability to generate an operational product from source or object code.

The installation, generation, and start-up procedures may exist as a separate document, but would typically be grouped with other administrative guidance.

4.2.2.1 ADO_IGS.1 Installation, Generation, and Start-up Procedures

Developer action elements:

~~(Req. 199)~~(Req. 204) ADO_IGS.1.1D: **The developer shall document procedures to be used for the secure installation, generation, and start-up of the product.**

Content and presentation of evidence elements:

~~(Req. 200)~~(Req. 205) ADO_IGS.1.1C: **The documentation shall describe the steps necessary for secure installation, generation, and start-up of the product.**

Evaluator action elements:

~~(Req. 201)~~(Req. 206) ADO_IGS.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.3 Development

4.3.1 Development ADV_FSP - Functional Specification

Objectives

The functional specification is a high-level description of the user-visible interface and behavior of the product. It is a refinement of the statement of functional requirements for the product. The functional specification must show that all defined functional requirements are addressed, and that the security policy is enforced by the product.

Application notes

In addition to the content indicated in the following requirements, the functional specification shall also include any additional specific detail specified by the documentation notes in the related functional products. For example, the functional specification shall contain the specification of the interaction (protocol) among various product products.

The developer must provide evidence that the product is completely represented by the functional specification. While a functional specification for the entire product would allow an evaluator to determine the product boundary, it is not necessary to require the specification of the boundary when other evidence could be provided to demonstrate the product boundary.

The evaluator of the product is expected to make determinations regarding the relevance of the functional specification to the functional requirements. In the course of the functional specification evaluation, there are essentially three types of evaluator determination: specific functional requirements are met and no further work (e.g., with a less abstract representation of the product) is necessary; specific functional requirements are violated and the product fails to meet its requirements; and

specific functional requirements have not been addressed and further analysis (of another product representation) is necessary. Whenever additional analysis is necessary, the evaluator is expected to carry that information forward to the analysis of other product representations. If requirements are not addressed after the analysis of the last provided product representation, this also represents a failure of the product evaluation.

In all cases, it is important that the evaluator evaluate the product as a unit since, in many cases, the security functions must cooperate to meet specific functional requirements, and each security function must not interfere with the operation of any other security function.

An informal security policy model can be a representation of the security policy in any notation, including a series of statements in the English Language.

4.3.1.1 ~~ADV_FSP.1~~ Functional Specification and Security Policy

Developer action elements:

~~(Req. 202)~~(Req. 207) ADV_FSP.1.1D: **The developer shall provide a functional specification.**

~~(Req. 203)~~(Req. 208) ADV_FSP.1.2D: **The developer shall provide a product security policy.**

Content and presentation of evidence elements:

~~(Req. 204)~~(Req. 209) ADV_FSP.1.1C: **The functional specification shall describe the product using an informal style.**

~~(Req. 205)~~(Req. 210) ADV_FSP.1.2C: **The functional specification shall include an informal presentation of syntax and semantics of all external product interfaces.**

~~(Req. 206)~~(Req. 211) ADV_FSP.1.3C: **The functional specification shall include evidence that demonstrates that the product is completely represented.**

Evaluator action elements:

~~(Req. 207)~~(Req. 212) ADV_FSP.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 208)~~(Req. 213) ADV_FSP.1.2E**: The evaluator shall determine that the functional specification is consistent with the product security policy.**

~~(Req. 209)~~(Req. 214) ADV_FSP.1.3E**: The evaluator shall determine if the functional requirements are addressed by the representation of the product, i.e., the functional specification.**

4.3.1.2 ADV_FSP.2 Functional Specification, Security Policy, and Informal Security Policy Model

Developer action elements:

~~(Req. 210)~~(Req. 215) ADV_FSP.2.1D: **The developer shall provide a functional specification.**

~~(Req. 211)~~(Req. 216) ADV_FSP.2.2D**: The developer shall provide a product security policy.**

~~(Req. 212)~~(Req. 217) ADV_FSP.2.3D**: The developer shall provide an informal security policy model.**

~~(Req. 213)~~(Req. 218) ADV_FSP.2.4D**: The developer shall provide a demonstration of the correspondence between the informal security policy model and the functional specification.**

Content and presentation of evidence elements:

~~(Req. 214)~~(Req. 219) ADV_FSP.2.1C: **The functional specification shall describe the product using an informal style.**

~~(Req. 215)~~(Req. 220) ADV_FSP.2.2C: **The functional specification shall include an informal presentation of the syntax and semantics of all external product interfaces.**

~~(Req. 216)~~(Req. 221) ADV_FSP.2.3C: **The functional specification shall include evidence that demonstrates that the product is completely represented.**

~~(Req. 217)~~(Req. 222) ADV_FSP.2.4C: **The demonstration of correspondence between the informal security policy model and the functional specification shall describe how the functional specification satisfies the informal security policy model.**

~~(Req. 218)~~(Req. 223) ADV_FSP.2.5C: **The demonstration of correspondence between the informal security policy model and the functional specification shall show that there are no security functions in the functional specification that conflict with the informal security policy model.**

~~(Req. 219)~~(Req. 224) ADV_FSP.2.6C: **The informal security policy model shall describe the rules and characteristics of all policies of the product that can be modeled.**

~~(Req. 220)~~(Req. 225) ADV_FSP.2.7C: **The informal security policy model shall include a rationale that demonstrates that policies that are modeled are satisfied by the informal security policy model.**

~~(Req. 221)~~(Req. 226) ADV_FSP.2.8C: **The informal security policy model shall justify that all policies that can be modeled are represented in the informal security policy model.**

Evaluator action elements:

~~(Req. 222)~~(Req. 227) ADV_FSP.2.1E**: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 223)~~(Req. 228) ADV_FSP.2.2E**: The evaluator shall determine that the functional specification is consistent with the product security policy.**

~~(Req. 224)~~(Req. 229) ADV_FSP.2.3E: **The evaluator shall determine if the functional requirements are addressed by the representation of the product, i.e., the functional specification.**

4.3.2 Development ADV_HLD - High-Level Design

Objectives

The high-level design of a product provides a description of the product in terms of major structural units (i.e., modules) and relates these units to the functions that they contain. The high-level design provides assurance that the product provides an architecture appropriate to implement the claimed functional requirements.

The high-level design refines the functional specification into modules. For each module of the product, the high-level design describes its purpose and function and identifies the security functions enforced by the module. The interrelationships of all modules are also defined in the high-level design. These interrelationships will be represented as external interfaces for data flow, control flow, etc., as appropriate.

Application notes

In addition to the content indicated in the following requirements,

~~(Req. 225)~~(Req. 230) The high-level design shall also include any additional specific detail specified by the documentation notes in the related functional products.

The developer is expected to describe the design of the product in terms of modules. The term ``module" is used here to express the idea of decomposing the product into a relatively small number of parts. While the developer is not required to actually have ``modules", the developer is expected to represent a similar level of decomposition. For example, a design may be similarly decomposed using ``layers", ``domains", or ``servers".

The evaluator of the product is expected to make determinations regarding the functional requirements in the product relevant to the high-level design. In the course of the high-level design evaluation, there are essentially three types of evaluator determination: specific functional requirements are met and no further work (e.g., with a less abstract representation of the product) is necessary; specific functional requirements are violated and the product fails to meet its requirements; and specific functional requirements have not been addressed and further analysis (of another product representation) is necessary. Whenever more analysis is necessary, the evaluator is expected to carry that information forward to the analysis of other product representations. If requirements are not addressed after the analysis of the last provided product representation, this also represents a failure of the product evaluation.

In all cases, it is important that the evaluator evaluate the product as a unit since in many cases the security functions must cooperate to meet specific functional requirements and also each security function must not interfere with the operation of any other security function.

The term ``security functionality'' is used to represent operations that a module performs that have some effect on the security functions implemented by the product. This distinction is made because modules do not necessarily relate to specific security functions. While a given module may correspond directly to a security function, or even multiple security functions, it is also possible that many modules must be combined to implement a single security function.

The term ``security policy enforcing modules'' refers to a module that contributes to the enforcement of the security policy.

4.3.2.1 ADV_HLD.1 Descriptive High-Level Design

Developer action elements :

~~(Req. 226)~~(Req. 231) ADV_HLD.1.1D: **The developer shall provide the high-level design of the product.**

Content and presentation of evidence elements:

~~(Req. 227)~~(Req. 232) ADV_HLD.1.1C: **The presentation of the high-level design shall be informal.**

~~(Req. 228)~~(Req. 233) ADV_HLD.1.2C: **The high-level design shall describe the structure of the product in terms of modules.**

~~(Req. 229)~~(Req. 234) ADV_HLD.1.3C: **The high-level design shall describe the security functionality provided by each module of the product.**

~~(Req. 230)~~(Req. 235) ADV_HLD.1.4C: **The high-level design shall identify the interfaces of the modules of the product.**

~~(Req. 231)~~(Req. 236) ADV_HLD.1.5C: **The high-level design shall identify any underlying hardware, firmware, and/or software required by the product with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.**

Evaluator action elements:

~~(Req. 232)~~(Req. 237) ADV_HLD.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 233)~~(Req. 238) ADV_HLD.1.2E: **The evaluator shall determine if the functional requirements in the product are addressed by the design.**

4.3.2.2 ADV_HLD.2 Security Enforcing High-Level Design

Developer action elements :

~~(Req. 234)~~(Req. 239) ADV_HLD.2.1D: **The developer shall provide the high-level design of the product.**

Content and presentation of evidence elements:

~~(Req. 235)~~(Req. 240) ADV_HLD.2.1C**: The presentation of the high-level design shall be informal.**

~~(Req. 236)~~(Req. 241) ADV_HLD.2.2C: **The high-level design shall describe the structure of the product in terms of modules.**

~~(Req. 237)~~(Req. 242) ADV_HLD.2.3C: **The high-level design shall describe the security functionality provided by each module of the product.**

~~(Req. 238)~~(Req. 243) ADV_HLD.2.4C: **The high-level design shall identify the interfaces of the modules of the product.**

~~(Req. 239)~~(Req. 244) ADV_HLD.2.5C: **The high-level design shall identify any underlying hardware, firmware, and/or software required by the product with a presentation of the functions provided by the supporting protection mechanisms implemented in that hardware, firmware, or software.**

~~(Req. 240)~~(Req. 245) ADV_HLD.2.6C: **The high-level design shall describe the separation of the product into security policy enforcing modules and other modules.**

Evaluator action elements:

~~(Req. 241)~~(Req. 246) ADV_HLD.2.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 242)~~(Req. 247) ADV_HLD.2.2E: **The evaluator shall determine if the functional requirements in the product are addressed by the design.**

4.3.3 Development ADV_IMP - Implementation Representation

Objectives

The description of the implementation in the form of source code, firmware, hardware drawings, etc. captures the detailed internal workings of the product in support of analysis.

Application notes

The implementation representation is used to express the notion of the least abstract representation of the product, specifically the one that is used to create the product itself without further design refinement. Source code which is then compiled or a hardware drawing which is used to build the actual hardware are examples of parts of an implementation representation.

The evaluator of the product is expected to make determinations regarding the functional requirements in the security target relevant to the implementation. In the course of the implementation evaluation, there are essentially three types of evaluator determination: specific functional requirements are met and no further work (e.g., with a more abstract representation of the product) is necessary; specific functional requirements are violated and the product fails to meet its requirements; and specific functional requirements have not been addressed and further analysis is necessary.

However, since the implementation is the least abstract representation it is likely that further analysis cannot be performed unless the product representations have not been evaluated in the usual order (i.e., most abstract to least abstract). If requirements are not addressed after the analysis of all product representations, this represents a failure of the product evaluation. Note that this more comprehensive failure determination requirement is realized in the Representation correspondence (ADV_RCR) family.

In all cases, it is important that the evaluator evaluates the product as a unit since, in many cases, the security functions must cooperate to meet specific functional requirements and each security function must not interfere with the operation of any other security function.

It is expected that evaluators will use the implementation to directly support other evaluation activities (e.g., vulnerability analysis, test coverage analysis).

4.3.3.1 ADV_IMP.1 Subset of the Implementation

Application notes

The implementation representation needs to be provided for the security relevant functions of the product. Any hardware, software, and/or firmware that does not contribute to the security need not be provided, analyzed, or tested. However, an explanation must be provided, and the evaluator must agree that the excluded items are not security relevant.

Developer action elements:

~~(Req. 243)~~(Req. 248) ADV_IMP.1.1D: **The developer shall provide the implementation representations for a selected subset of the product.**

Content and presentation of evidence elements:

~~(Req. 244)~~(Req. 249) ADV_IMP.1.1C: **The implementation representations shall unambiguously define the product to a level of detail such that it can be generated without further design decisions.**

Evaluator action elements:

~~(Req. 245)~~(Req. 250) ADV_IMP.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 246)~~(Req. 251) ADV_IMP.1.2E: **The evaluator shall determine if the KRS functional requirements are addressed by the representation of the product. [NOTE: DOES THIS MEAN THAT THE FUNCTIONAL REQUIREMENTS NEED TO BE IDENTIFIED AS SUCH SO THAT THERE IS NO CONFUSION WITH SECURITY OR OPERATIONAL REQUIREMENTS?]**

4.3.4 Development ADV_LLD - Low-Level Design

Objectives

The low-level design of a product provides a description of the internal workings of the product in terms of modules and their interrelationships and dependencies. The low-level design provides assurance that the modules have been correctly and effectively refined.

For each module of the product, the low-level design describes its purpose, function, interfaces, dependencies, and the implementation of any security policy enforcing functions.

Application notes
In addition to the content indicated in the following requirements, the low-level design shall also include any additional specific detail specified by the documentation notes in the related functional products.

The evaluator of the product is expected to make determinations regarding the functional requirements relevant to the low-level design. In the course of the low-level design evaluation, there are essentially three types of evaluator determination: specific functional requirements are met and no further work (e.g., with a less abstract representation of the product) is necessary; specific functional requirements are violated and the product fails to meet its requirements; and specific functional requirements have not been addressed and further analysis (of another product representation) is necessary. Whenever more analysis is necessary, the evaluator is expected to carry that information forward to the analysis of other product representations. If requirements are not addressed after the analysis of the last provided product representation, this also represents a failure of the product evaluation. Note that this more comprehensive failure determination requirement is realized in the Representation correspondence (ADV_RCR) family.

In all cases, it is important that the evaluator evaluates the product as a unit since, in many cases, the security functions must cooperate to meet specific functional requirements, and each security function must not interfere with the operation of any other security function.

4.3.4.1 ADV_LLD.1 Descriptive Low-Level Design

Application notes
Only representations for modules in the product need to be provided.

Developer action elements:

~~(Req. 247)~~(Req. 252) ADV_LLD.1.1D: **The developer shall provide the low-level design of the product.**

Content and presentation of evidence elements:

~~(Req. 248)~~(Req. 253) ADV_LLD.1.1C: **The presentation of the low-level design shall be informal.**

~~(Req. 249)~~(Req. 254) ADV_LLD.1.2C: **The low-level design shall describe the product in terms of modules.**

~~(Req. 250)~~(Req. 255) ADV_LLD.1.3C: **The low-level design shall describe the purpose of each module.**

~~(Req. 251)~~(Req. 256) ADV_LLD.1.4C: **The low-level design shall define the interrelationships between the modules in terms of provided functionality and dependencies on other modules.**

~~(Req. 252)~~(Req. 257) ADV_LLD.1.5C: **The low-level design shall describe the implementation of all security policy enforcing functions.**

~~(Req. 253)~~(Req. 258) ADV_LLD.1.6C: **The low-level design shall describe the interfaces of each module in terms of their syntax and semantics.**

~~(Req. 254)~~(Req. 259) ADV_LLD.1.7C: **The low-level design shall provide a demonstration that the product is completely represented.**

~~(Req. 255)~~(Req. 260) ADV_LLD.1.8C: **The low-level design shall identify the interfaces of the modules of the product which are visible at the external interface of the product.**

Evaluator action elements:

~~(Req. 256)~~(Req. 261) ADV_LLD.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 257)~~(Req. 262) ADV_LLD.1.2E: **The evaluator shall determine if the functional requirements in the KRS are addressed by the representation of the product.**

4.3.5 Development ADV_RCR - Representation Correspondence

Objectives

The correspondence between the various representations (i.e. functional requirements expressed in the KRS, functional specification, high-level design, low-level design, implementation) addresses the correct and complete instantiation of the requirements to the least abstract representation provided. This conclusion is achieved by step-wise refinement and the cumulative results of correspondence determinations between all adjacent abstractions of representation.

Application notes

The developer must demonstrate to the evaluator that the most detailed, or least abstract, representation of the product is an accurate, consistent, and complete instantiation of the functions expressed as functional requirements in this standard. This is accomplished by showing correspondence between adjacent representations at a commensurate level of rigor.

The evaluator must analyze each demonstration of correspondence between abstractions, as well as the results of the analysis of each product representation, and then make a determination as to whether the functional requirements in this standard have been satisfied.

This family of requirements is not intended to address correspondence relating to the security policy model. Rather, it is intended to address correspondence between the requirements in this standard as well as the product, functional specification, high-level design, low-level design, and implementation representation.

4.3.5.1 ADV_RCR.1 Informal Correspondence Demonstration

Developer action elements:

~~(Req. 258)~~(Req. 263) ADV_RCR.1.1D: **The developer shall provide evidence that the least abstract product representation provided is an accurate, consistent, and complete instantiation of the functional requirements expressed in this standard.**

Content and presentation of evidence elements:

~~(Req. 259)~~(Req. 264) ADV_RCR.1.1C: **For each adjacent pair of product representations, the evidence shall demonstrate that all parts of the more abstract representation are refined in the less abstract representation.**

~~(Req. 260)~~(Req. 265) ADV_RCR.1.2C: **For each adjacent pair of product representations, the demonstration of correspondence between the representations may be informal.**

Evaluator action elements:

~~(Req. 261)~~(Req. 266) ADV_RCR.1.1E**: The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 262)~~(Req. 267) ADV_RCR.1.2E: **The evaluator shall analyze the correspondence between the functional requirements expressed in this standard and the least abstract representation provided by the developer in order to ensure accuracy, consistency, and completeness.**

4.4 Guidance Documents

4.4.1 Guidance Documents AGD_ADM Administrator Guidance

Objectives

Administrator guidance refers to written material that is intended to be used by those persons responsible for configuring, maintaining, and administering the product in a correct manner for maximum security. Because the secure operation of the product is dependent upon the correct performance of the product, persons responsible for performing these functions are trusted by the product. Administrator guidance is intended to help administrators understand the security functions provided by the product, including both those functions that require the administrator to perform security-critical actions and those functions that provide security-critical information.

Application notes

The requirements AGD_ADM.1.2C and AGD_ADM.1.11C encompass the aspect that any warnings to the users of a product with regard to the product security environment and the security objectives described in this standard are appropriately covered in the administrator guidance.

Those topics that are relevant to administrator guidance for the understanding and proper application of the security functions should be considered for inclusion in the administrator guidance requirements. An example of an administrator guidance document is a reference manual.

4.4.1.1 AGD_ADM.1 Administrator Guidance

Developer action elements:

~~(Req. 263)~~(Req. 268) AGD_ADM.1.1D: **The developer shall provide administrator guidance addressed to system administrative personnel.**

Content and presentation of evidence elements:

~~(Req. 264)~~(Req. 269) AGD_ADM.1.1C: **The administrator guidance shall describe how to administer the product in a secure manner.**

~~(Req. 265)~~(Req. 270) AGD_ADM.1.2C: **The administrator guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.**

~~(Req. 266)~~(Req. 271) AGD_ADM.1.3C: **The administrator guidance shall contain guidelines on the consistent and effective use of the security functions within the product.**

~~(Req. 267)~~(Req. 272) AGD_ADM.1.4C: **The administrator guidance shall describe the difference between two types of functions: those which allow an administrator to control security parameters, and those which allow the administrator to obtain information only.**

~~(Req. 268)~~(Req. 273) AGD_ADM.1.5C: **The administrator guidance shall describe all security parameters under the administrator's control.**

~~(Req. 269)~~(Req. 274) AGD_ADM.1.6C: **The administrator guidance shall describe each type of security-relevant event relative to the administrative functions that need to be performed, including changing the security characteristics of entities under the control of the product.**

~~(Req. 270)~~(Req. 275) AGD_ADM.1.7C: **The administrator guidance shall contain guidelines on how the security functions interact.**

~~(Req. 271)~~(Req. 276) AGD_ADM.1.8C: **The administrator guidance shall contain instructions regarding how to configure the product.**

~~(Req. 272)~~(Req. 277) AGD_ADM.1.9C: **The administrator guidance shall describe all configuration options that may be used during the secure installation of the product.**

~~(Req. 273)~~(Req. 278) AGD_ADM.1.10C: **The administrator guidance shall describe details, sufficient for the use of procedures relevant to the administration of security.**

~~(Req. 274)~~(Req. 279) AGD_ADM.1.11C: **The administrator guidance shall be consistent with all other documents supplied for evaluation.**

Evaluator action elements:

~~(Req. 275)~~(Req. 280) AGD_ADM.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 276)~~(Req. 281) AGD_ADM.1.2E: **The evaluator shall confirm that the installation procedures result in a secure configuration.**

4.4.2 Guidance Documents AGD_USR - User Guidance

Objectives

User guidance refers to written material that is intended to be used by non-administrative (human) users of the product. User guidance describes the security functions provided by the product and provides instructions and guidelines, including warnings, for its secure use.

The user guidance provides a basis for assumptions about the use of the product and a measure of confidence that non-malicious users and application providers will understand the secure operation of the product and will use it as intended.

Application notes

The requirement AGD_USR.1.3.C and AGD_USR.1.5C encompass the aspect that any warnings to the users of a product with regard to the product security environment and the security objectives described in this standard are appropriately covered in the user guidance.

Those topics in this standard that are relevant to user guidance aimed at the understanding and proper use of the security functions should be considered for inclusion in the user guidance requirements. Examples of user guidance are reference manuals, user guides, and on-line help.

4.4.2.1 AGD_USR.1 User Guidance

Developer action elements:

~~(Req. 277)~~(Req. 282) AGD_USR.1.1D: **The developer shall provide user guidance.**

Content and presentation of evidence elements:

~~(Req. 278)~~(Req. 283) AGD_USR.1.1C: **The user guidance shall describe the product and interfaces available to the user.**

~~(Req. 279)~~(Req. 284) AGD_USR.1.2C: **The user guidance shall contain guidelines on the use of security functions provided by the product.**

~~(Req. 280)~~(Req. 285) AGD_USR.1.3C: **The user guidance shall contain warnings about functions and privileges that should be controlled in a secure processing environment.**

~~(Req. 281)~~(Req. 286) AGD_USR.1.4C: **The user guidance shall describe the interaction between user-visible security functions.**

~~(Req. 282)~~(Req. 287) AGD_USR.1.5C: **The user guidance shall be consistent with all other documentation delivered for evaluation.**

Evaluator action elements:

~~(Req. 283)~~(Req. 288) AGD_USR.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.5 Life Cycle Support

4.5.1 Life Cycle Support ALC_FLR - Flaw Remediation

Objectives

Flaw remediation requires that discovered flaws be tracked and corrected by the developer. Although future compliance with flaw remediation procedures cannot be determined at the time of the product evaluation, it is possible to evaluate the policies and procedures that a developer has in place to track and correct flaws, and to distribute the flaw information and corrections.

Application notes

None

4.5.1.1 ALC_FLR.1 Basic Flaw Remediation

Developer action elements:

~~(Req. 284)~~(Req. 289) ALC_FLR.1.1D: **The developer shall document the flaw remediation procedures.**

Content and presentation of evidence elements:

~~(Req. 285)~~(Req. 290) ALC_FLR.1.1C: **The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the product.**

~~(Req. 286)~~(Req. 291) ALC_FLR.1.2C: **The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.**

~~(Req. 287)~~(Req. 292) ALC_FLR.1.3C: **The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.**

~~(Req. 288)~~(Req. 293) ALC_FLR.1.4C: **The flaw remediation procedures documentation shall describe the methods used to provide flaw information and corrections to product users.**

Evaluator action elements:

~~(Req. 289)~~(Req. 294) ALC_FLR.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.5.1.2 ALC_FLR.2 Flaw Reporting Procedures

Developer action elements:

~~(Req. 290)~~(Req. 295) ALC_FLR.2.1D: **The developer shall document the flaw remediation procedures.**

~~(Req. 291)~~(Req. 296) ALC_FLR.2.2D: **The developer shall establish a procedure for accepting and acting upon user reports of security flaws and requests for corrections to those flaws.**

Content and presentation of evidence elements:

~~(Req. 292)~~(Req. 297) ALC_FLR.2.1C: **The flaw remediation procedures documentation shall describe the procedures used to track all reported security flaws in each release of the product.**

~~(Req. 293)~~(Req. 298) ALC_FLR.2.2C: **The flaw remediation procedures shall require that a description of the nature and effect of each security flaw be provided, as well as the status of finding a correction to that flaw.**

~~(Req. 294)~~(Req. 299) ALC_FLR.2.3C: **The flaw remediation procedures shall require that corrective actions be identified for each of the security flaws.**

~~(Req. 295)~~(Req. 300) ALC_FLR.2.4C: **The flaw remediation procedures documentation shall describe the methods used to provide flaw information and corrections to product users.**

~~(Req. 296)~~(Req. 301) ALC_FLR.2.5C: **The procedures for processing reported security flaws shall ensure that any reported flaws are corrected and the correction issued to product users.**

Evaluator action elements:

~~(Req. 297)~~(Req. 302) ALC_FLR.2.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.6 Tests

4.6.1 Tests ATE_COV - Coverage

Objectives
This family addresses those aspects of testing that deal with completeness of testing. That is, it addresses the extent to which the product security functions are tested, whether or not the testing is sufficiently extensive to demonstrate that the product operates as specified, and whether or not the order in which testing proceeds correctly accounts for functional dependencies between the portions of the product being tested.

Application notes
The specific documentation required by the coverage products will be determined, in most cases, by the documentation stipulated in the level of ATE_FUN that is specified.

4.6.1.1 ATE_COV.1 Complete Coverage - Informal

Objectives

In this component, the objective is that testing completely address the security functions.

Application notes
While the testing objective is to completely cover the product, there is no more than an informal explanation to support this assertion.

Developer action elements:

~~(Req. 298)~~(Req. 303) ATE_COV.1.1D: **The developer shall provide an analysis of the test coverage.**

Content and presentation of evidence elements:

~~(Req. 299)~~(Req. 304) ATE_COV.1.1C: **The analysis of the test coverage shall demonstrate that the tests identified in the test documentation cover the product.**

Evaluator action elements:

~~(Req. 300)~~(Req. 305) ATE_COV.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.6.2 Tests ATE_DPT - Depth

Objectives
The products in this family deal with the level of detail to which the product is tested. The testing of security functions is based upon an increasing depth of information derived from the analysis of the representations.

The objective is to counter the risk of missing an error in the development of the product. Additionally, the products of this family, especially as testing is more concerned with the internals of the product, are more likely to discover any malicious code that has been inserted.

Application notes
The specific amount and type of documentation and evidence will, in general, be determined by that required by the level of ATE_FUN selected.

4.6.2.1 ATE_DPT.1 Testing - Functional Specification

Objectives

The functional specification of a product provides a high level description of the external workings of the product. Testing at the level of the functional specification, in order to demonstrate the presence of any flaws, provides assurance that the product functional specification has been correctly realized.

Application notes

The functional specification representation is used to express the notion of the most abstract representation of the product.

Developer action elements:

~~(Req. 301)~~(Req. 306) ATE_DPT.1.1D: **The developer shall provide the analysis of the depth of testing.**

Content and presentation of evidence elements:

~~(Req. 302)~~(Req. 307) ATE_DPT.1.1C: **The depth analysis shall demonstrate that the tests identified in the test documentation are sufficient to demonstrate that the product operates in accordance with the functional specification of the product.**

Evaluator action elements:

~~(Req. 303)~~(Req. 308) ATE_DPT.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.6.3 Tests ATE_FUN - Functional Tests

Objectives

Functional testing establishes that the product exhibits the properties necessary to satisfy the functional requirements of this standard. Functional testing provides assurance that the product satisfies at least the security functional requirements, although it cannot establish that the product does no more than what was specified. The ``Functional tests" family is focused on the type and amount of documentation or support tools required, and what is to be demonstrated through testing.

This family contributes to providing assurance that the likelihood of undiscovered flaws is relatively small.

Application notes
Procedures for performing tests are expected to provide instructions for using test programs and test suites, including the test environment, test conditions, test data parameters and values. The test procedures should also show how the test results are derived from the test inputs.

The developer shall eliminate all security relevant flaws discovered during testing.

The developer shall test the product to determine that no new security relevant flaws have been introduced as a result of eliminating discovered security relevant flaws.

Tests shall include an examination of procedures and documents that assist in implementing the product security policy.

4.6.3.1 ATE_FUN.1 Functional Testing

Objectives
The objective is for the developer to demonstrate that all security functions perform as specified. The developer is required to perform testing and to provide test documentation.

Developer action elements:

~~(Req. 304)~~(Req. 309) ATE_FUN.1.1D: **The developer shall test the product and document the results.**

~~(Req. 305)~~(Req. 310) ATE_FUN.1.2D: **The developer shall provide test documentation.**

Content and presentation of evidence elements:

~~(Req. 306)~~(Req. 311) ATE_FUN.1.1C: **The test documentation shall consist of test plans, test procedure descriptions, and test results.**

~~(Req. 307)~~(Req. 312) ATE_FUN.1.2C: **The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed.**

~~(Req. 308)~~(Req. 313) ATE_FUN.1.3C: **The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function.**

~~(Req. 309)~~(Req. 314) ATE_FUN.1.4C: **The test results in the test documentation shall show the expected results of each test.**

~~(Req. 310)~~(Req. 315) ATE_FUN.1.5C: **The test results from the execution of the tests by the developer shall demonstrate that each security function operates as specified.**

Evaluator action elements:

~~(Req. 311)~~(Req. 316) ATE_FUN.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

4.6.4 Tests ATE_IND - Independent Testing

Objectives

The objective is to demonstrate that the security functions perform as specified.

An additional objective is to counter the risk of an incorrect assessment of the test outcomes on the part of the developer which results in the incorrect implementation of the specifications, or overlooks code that is non-compliant with the specifications.

Application notes

The testing specified in this family can be performed by a party other than the evaluator (e.g., an independent laboratory, an objective consumer organization).

This family deals with the degree to which there is independent functional testing of the product. Independent functional testing may take the form of repeating the developer's functional tests in whole or in part. It may also take the form of the augmentation of the developer's functional tests, either to extend the scope or the depth of the developer's tests.

Independent testing shall be performed by an independent third party certified and accredited by the Government.

The Government will supply some tests to validate compliance and conformance. Examples include: cryptographic algorithms and cryptographic protocols. The evaluator (which happens to be the independent third party) shall execute these government supplied tests in addition to the tests provided by the developer, and tests developed by the evaluator.

4.6.4.1 ATE_IND.2 Independent Testing - Sample

Objectives
The objective is to demonstrate that the security functions perform as specified.

In this component, the objective is to select and repeat a sample of the developer testing.

Application notes
The suitability of the product for testing is based on access to the product, and the supporting documentation and information required to run tests. The need for documentation is supported by other assurance families (e.g., ATE_FUN)

Additionally, the suitability of the product for testing may be based on other considerations (e.g., the version of the product submitted by the developer is not the final version).

The developer is required to perform testing and to provide test documentation and test results. This is addressed by the ATE_FUN family.

Testing may be selective and is based upon all available documentation.

Developer action elements:

~~(Req. 312)~~(Req. 317) ATE_IND.2.1D: **The developer shall provide the product for testing.**

Content and presentation of evidence elements:

~~(Req. 313)~~(Req. 318) ATE_IND.2.1C: **The product shall be suitable for testing.**

Evaluator action elements:

~~(Req. 314)~~(Req. 319) ATE_IND.2.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 315)~~(Req. 320) ATE_IND.2.2E: **The evaluator shall test the product to confirm that the product operates as specified.**

~~(Req. 316)~~(Req. 321) ATE_IND.2.3E: **The evaluator shall execute a sample of tests in the test documentation to verify the developer test results.**

4.6.4.2 ATE_IND.3 Independent Testing - Complete

Objectives
The objective is to demonstrate that the security functions perform as specified.

In this component, the objective is to repeat the developer testing.

Application notes
The suitability of the product for testing is based on access to the product, and the supporting documentation and information required to run tests. The need for documentation is supported by other assurance families (e.g., ATE_FUN)

Additionally, the suitability of the product for testing may be based on other considerations (e.g., the version of the product submitted by the developer is not the final version).

The developer is required to perform testing and to provide test documentation and test results. This is addressed by the ATE_FUN family.

Developer action elements:

~~(Req. 317)~~(Req. 322) ATE_IND.3.1D: **The developer shall provide the product for testing.**

Content and presentation of evidence elements:

~~(Req. 318)~~(Req. 323) ATE_IND.3.1C: **The product shall be suitable for testing.**

Evaluator action elements:

~~(Req. 319)~~(Req. 324) ATE_IND.3.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 320)~~(Req. 325) ATE_IND.3.2E: **The evaluator shall test the product to confirm that the product operates as specified.**

~~(Req. 321)~~(Req. 326) ATE_IND.3.3E: **The evaluator shall execute all tests in the test documentation to verify the developer test results.**

4.7 Vulnerability Assessment

4.7.1 Vulnerability Assessment AVA_VLA - Vulnerability Analysis

Objectives

Vulnerability analysis is an assessment to determine whether vulnerabilities could allow malicious users to violate the security policy. These vulnerabilities will be identified during the evaluation by flaw hypotheses.

Vulnerability analysis deals with the threats that a malicious user will be able to discover flaws that will allow access to resources (e.g., data), allow the ability to interfere with or alter the product, or interfere with the authorized capabilities of other users.

Application notes

The vulnerability analysis should consider the contents of all the product deliverables for the targeted evaluation assurance level.

Obvious vulnerabilities are those that allow common attacks or those that might be suggested by the product interface description. Obvious vulnerabilities are those in the public domain, details of which should be known to a developer, publicly available, or available from NIST.

The evidence identifies all the product documentation upon which the search for flaws was based.

4.7.1.1 AVA_VLA.1 Developer Vulnerability Analysis

Objectives

A vulnerability analysis is performed by the developer to ascertain the presence of ``obvious'' security vulnerabilities.

The objective is to confirm that no identified security vulnerabilities can be exploited in the intended environment for the product.

Application notes

Obvious vulnerabilities are those which are open to exploitations which require a minimum of understanding of the product, skill, technical sophistication, and resources.

Developer action elements:

~~(Req. 322)~~(Req. 327) AVA_VLA.1.1D: **The developer shall perform and document an analysis of the product deliverables searching for obvious ways in which a user can violate the security policy.**

~~(Req. 323)~~(Req. 328) AVA_VLA.1.2D: **The developer shall document the disposition of identified vulnerabilities.**

Content and presentation of evidence elements:

~~(Req. 324)~~(Req. 329) AVA_VLA.1.1C: **The evidence shall show, for each vulnerability, that the vulnerability cannot be exploited in the intended environment for the product.**

Evaluator action elements:

~~(Req. 325)~~(Req. 330) AVA_VLA.1.1E: **The evaluator shall confirm that the information provided meets all requirements for content and presentation of evidence.**

~~(Req. 326)~~(Req. 331) AVA_VLA.1.2E: **The evaluator shall conduct penetration testing, based on the developer vulnerability analysis, to ensure that obvious vulnerabilities have been addressed.**

4.8 Excluded Assurance Requirements

This section contains the Common Criteria assurance requirements that are recommended for exclusion.

The ADV_INT family relates to modularity, layering, information hiding, etc. For economic reasons, this family has not been included.

ALC_DVS (Developmental Security), ALC_LCD (Life Cycle Definition), and ALC_TAT (Tools and Techniques) are excluded in order to provide engineering independence for the vendors, spur commercial product development, and align assurance requirements with the commercial practices.

AVA_CCA (Covert Channel Analysis), AVA_SOF (Strength of Function, e.g., work factor for cryptographic operation) are excluded since they are not particularly relevant here. AVA_CCA in non-discretionary policy environments can be implemented using procedural controls such as executing trusted software only. Cryptanalysis work factors will be provided or implied by the FIPS cryptographic algorithms.

AVA_MSU (Misuse Analysis) is excluded since obvious flaws and known flaws will come under AVA_VLA (Vulnerability Analysis). Given this is a standard for SBU data, vulnerability analysis may be an overkill.

~~5 Key Recovery Requestor to Key Recovery Agent Syntax~~

~~5.1 Key Recovery Request~~

~~(Req. 327)A key recovery request body shall include the following:~~

- ~~originator identity~~
- ~~recipient identity~~
- ~~current date and time~~
- ~~date and time of encrypted key/data capture~~
- ~~key recovery block~~
- ~~optional: subject passphrase~~
- ~~optional: MIME formatted encrypted data~~

~~(Req. 328)Individual items within the request transaction body shall be delimited by blank lines.~~

- ~~Originator-id: <identifying information>~~
- ~~Recipient-id: <identifying information>~~
- ~~Date-Time: <date/time string>~~
- ~~Capture-Time: <date/time string>~~
- ~~KRB: <printably encoded recovery block>~~
- ~~Passphrase: <subject authentication data>~~
- ~~Encrypted-Data: <MIME encoded data>~~

~~TBD: BNF For Key Recovery Request~~

~~5.2 Key Recovery Response~~

~~(Req. 329)A key recovery response body shall include the following:~~

- ~~KRA identity~~
- ~~KRR identity~~
- ~~recipient identity~~
- ~~current date and time~~
- ~~date and time of key/data recovery~~
- ~~printably encoded key~~
- ~~optional: MIME formatted data~~

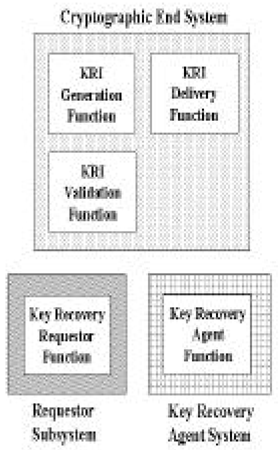
~~**(Req. 330)Individual items within the response transaction body shall be delimited by blank lines.**~~

~~**KRA-id: <**identifying information>~~
~~**KRR-id: <**identifying information>~~
~~**Recipient-id: <**identifying information>~~
~~**Date-Time:** <date/time string>~~
~~**Recovery-Time:** <date/time string>~~
~~**DEK:** <printably encoded encryption key>~~
~~**Data:** <MIME encoded data>~~

~~**TBD: BNF For Key Recovery Response**~~

Appendix A: Examples

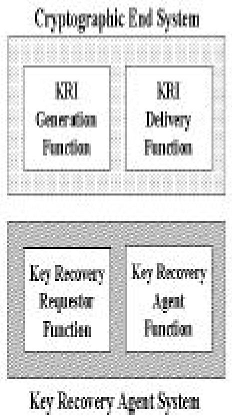
A.1 Key



integrated into products in a variety of configurations in order to accommodate different user environments. In Figure 4, the KRI Generation, Delivery, and Validation functions are provided in a single cryptographic end system product. The Requestor and KRA functions are each available as independent products. The separate Requestor System might be appropriate in an organization which prefers to centralize the key recovery process.

In Figure 5, the KRI Generation and Delivery Functions are provided in one product, while the Requestor Function and KRA Function are in a separate product. This configuration may be appropriate for a storage application, where files are encrypted by a user, KRI is attached to the file and thereafter ignored unless the decryption key becomes unavailable and recovery is required. The user could then go to a special recovery system in order to recover the appropriate key.

Figure 4


Cryptographic End System
KRI Generation Function
KRI Delivery Function
Key Recovery Requestor Function
Key Recovery Agent Function
Key Recovery Agent System

In Figure 6, the KRA function is bundled with the KRI Generation and Delivery Functions. This might be appropriate for an environment in which the KRA generates the encryption key pair, sends it off to the user and/or a CA for certification, and caches a copy of the private key for potential recovery at a later time.

In Figure 7, the KRI Generation, Delivery, Validation and Requestor Functions are provided in a single cryptographic end system. The KRA

Function is a separate product. There may be an electronic connection between the end user system and the KRA in order to effect the recovery process.

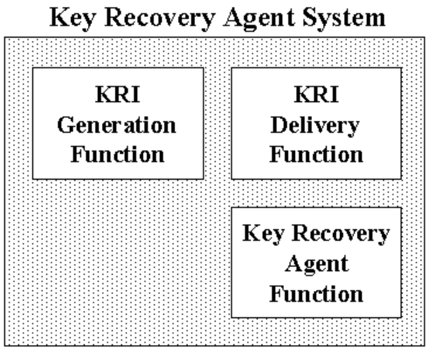


Figure 6

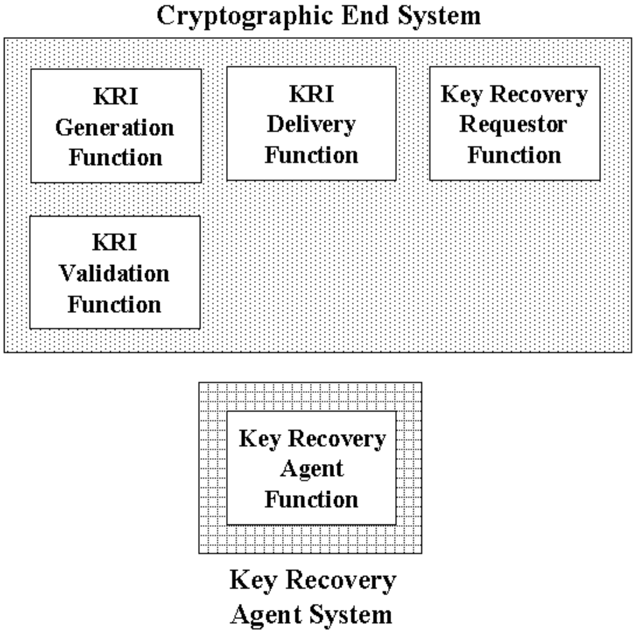


Figure 7

A.2 Multiple KRI Generation Functions

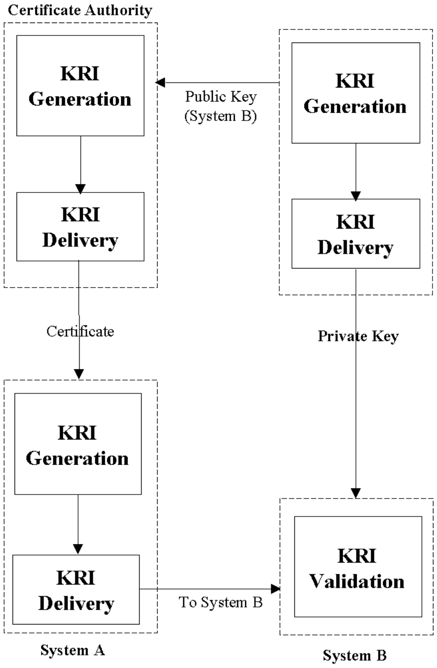


Figure 8 provides an example of multiple KRI Generation Functions which are required to provide the aggregate of KRI needed to recover a target key. Suppose that System B or a trusted generation service generates an encryption key pair for System B and provides the public key to a Certificate Authority (CA) along with other information which will be useful in providing key recovery. The CA generates a certificate containing this information. System A uses this certificate along with other internally generated information to create KRI for messages to be sent to System B. In this case, System A, the CA and whoever generates System B’s key pair participate in the generation of the KRI that will allow System B to recover.

A.3 KRI Generation Scenarios

Assume that each system has an encryption public key certificate (hereafter called an encryption certificate) that identifies the key recovery method and the identity of the KRA(s). Encryption certificates are also available for the KRAs.

Figure 8: Multiple KRI Generation Functions

A.3.1 ~~Interactive~~ Realtime Communications

A.3.1.1 Between Two Encapsulation Techniques

In Figure 10 (Appendix E), cryptographic end systems A and B are two systems that employ two different encapsulation methods for key recovery, but use a common key recovery block (KRB). A key transport method of key exchange is used (e.g., the data key is encrypted using the receiver's encryption public key). System A has a key recovery policy stating that key recovery information is not created for interactive communications. System B has a key recovery policy that states: (1) key recovery information must be created for itself for all communications when that information is not present, and (2) key recovery information must also be created for the other party whenever possible.

System A creates a data key to be used for the communication session and encrypts the data key using the public encryption key of System B (obtained from System B's encryption certificate). System A sends the encrypted key as part of the normal key exchange process. System A then encrypts a message for System B, and sends the encrypted message on the communications path.

When System B determines that no key recovery information is available for the message received from System A (i.e., no KRB is present), System B decrypts the encrypted data key (received as part of the key exchange process), and uses the resulting plaintext data key to create key recovery information for itself and/or its Key Recovery Agent. The KRI is placed in a KRB in accordance with its key recovery scheme. By examining System A's certificate, the identity of System A's KRA(s) can be determined, and the KRA encryption certificate(s) can be acquired. If System B can create a KRB for System A's key recovery technique and all information is available, key recovery information is created for System A and/or its Key Recovery Agent(s). System B then uses the data key to decrypt the received message. The newly created key recovery information is then attached to the next message in the communication session and sent back to System A.

In subsequent messages received by System A within this interactive session, System A can recognize the presence of the KRI (perhaps perform some processing of the KRI in the KRB) and decrypt the received messages.

A.3.1.2 Between Encapsulated and Key Escrow Techniques

Figure 12 (Appendix E) includes cryptographic end systems A and B that use key escrow and ~~KRI~~ key encapsulation methods of key recovery, respectively. System B uses a KRB. A key agreement method of key exchange is used (e.g., the encryption public and private keys pairs of both parties to a communication are used along with randomly generated values to generate a shared data key at the cryptographic end systems). System A has a key recovery policy that requires that all incoming communications must have KRI available for the sender. System B has a policy stating that communications will only be conducted with other parties that employ key recovery techniques, and that KRI is always created for itself in outgoing communications.

System B wants to initiate a communication session with System A. By obtaining System A's encryption certificate, System B obtains System A's public encryption key as well as determining that System A uses a key escrow method of key recovery. System B initiates a key exchange with System A to agree upon a data key, then encapsulates the data key and other KRI in a KRB for itself and its KRA. The data key is then used to encrypt the data, and the encrypted data and the KRB are sent to system A.

System A (probably during the key exchange process) determines that System B uses an encapsulated method of key recovery by examining System B's encryption certificate. When the initial message is received from System B, System A is able to recognize that there is a KRB for System B. System A then proceeds to decrypt the received message.

A.3.2 ~~Store and Forward~~Staged Delivery Communications

A.3.2.1 Between Two Key Escrow Key Recovery Schemes

In Figure 10 (Appendix E), cryptographic end Systems A and B employ key escrow methods of key recovery. A key transport method of key exchange is used. System B has a policy stating that all outgoing email messages will be archived and recoverable (i.e., KRI must be available to recover encrypted email messages that have been archived). System A is able to recover incoming encrypted email messages if key transport is used for key exchange.

System B generates a data key and encrypts the key using the encryption public key of the receiver SA) for use in the key exchange (key transport process). Even though System B uses key escrow, there is nothing yet which allows System B to recover after the outgoing message is archived. System B encrypts the data key using his own encryption public key, and places it in a KRB. System B then encrypts the message with the data key, and sends the encrypted message and System A's copy of the encrypted data key to System A. The encrypted message and the KRB are archived.

System A decrypts the data key received via the key transport mechanism and decrypts the received message using that key. [make gender neutral]

A.3.2.2 Between an Encapsulated Scheme and an End User System with No Key Recovery Capability

In Figure 9 (Appendix E), cryptographic end System A uses an encapsulated method of key recovery. System B has no key recovery capability. A key transport method of key exchange is in use (e.g., the data key is encrypted by the receiver's encryption public key). System A has a key recovery policy that states: (1) key recovery information must always be created for itself and/or its Key Recovery Agent, and (2) Key recovery Information is not created for anyone else. System A retains a copy of all outgoing email messages. System A sends the KRB along an alternate path from that of the encrypted messages; this allows system B to ignore key recovery information so that interoperability is possible.

System A creates a data key, then creates key recovery information for itself and/or its Key Recovery Agent, and places the KRI in a KRB. The KRB is sent along the alternate communication path. The data key is encrypted by system B's encryption public key (obtained from System B's encryption certificate) and then used to encrypt an e-mail message. The encrypted key is placed in the message header (the method of key transport that is employed in this example) and sent with the encrypted message to System B.

Upon receipt of the encrypted message and key exchange information , System B decrypts the data encryption key in the message header, and uses the decrypted data encryption key to decrypt the message.

A.3.3 Data Storage

A.3.3.1 Creation by an End User with an Encapsulated Scheme; Read Access by Anyone

For data storage applications, the Encryptor and Decryptor may not be the same entity (e.g., shared files). In Figure 9 (Appendix E), end user system A uses an encapsulated method for key recovery. System A's organization has a policy stating that key recovery information must exist for all stored data. Read only access can be granted to a list of other systems in the organization, whether or not those systems have a key recovery capability.

System A creates a data key and uses the encryption public key of each system on the access list to encrypt a copy of the data key for that system (including itself). System A also encrypts the data key using the encryption public key of the organization's KRA. The data key is then used to encrypt the data. All copies of the encrypted key are placed in a file along with the encrypted data.

When accessing the encrypted file, the acquiring system decrypts the appropriate copy of the encrypted data key, and uses the decrypted data key to decrypt the file.

A.4 Key Recovery Scenarios

A.4.1 ~~Interactive~~ Realtime Session

Referring back to scenario A.3.1.2, when System A initially tries to participate in the key exchange process, it is discovered that the private key of the encryption public key pair is lost. System A immediately requests the recovery of its private key from the KRA using its automated ability to request key recovery. When the private key is provided, system A can continue with the key exchange process and participate in the determination of the data key to be used for the communication session. [not credible, i.e., recovery scenario too rapid!]

A.4.2 ~~Store-and-Forward~~Staged Delivery Communications

In scenario A.3.2.1, the email message received by System A is stored in the in-box until read. Suppose that the user receives a large number of email messages before reading them. When attempting to read the encrypted messages, it is discovered that the private key of the encryption public key pair is corrupted. The user requests a recovery of the private key from the key recovery function, uses the recovered private key to decrypt the data key for each message, and then uses the data key to decrypt the associated message.

A.4.3 Data Storage

In scenario A.3.3.1, System A could create a file for himself (i.e., no one else is on the access list, so the data key is not encrypted for anyone else). At some later time, the user needs to retrieve the file, but has lost access to his decryption key. The data key can be recovered by sending the copy of the key which was encrypted using the KRA's encryption public key to the KRA for decryption. [make gender neutral]

Appendix B: Key Recovery Block

B.1 ~~Introduction~~Overview

When different key recovery products that employ ~~KRI~~ key encapsulation need to interoperate with one another, one of the major obstacles is the inability of the receiver product to recognize and validate the key recovery information received from the sender product. In order to allow the interoperability of various key recovery techniques which require the use of KRI encapsulation, a common structure -- a Key Recovery Block (KRB) -- may be required. The KRB serves as a container[12] for technique-specific key recovery information, and supports generic mechanisms to identify and validate the contained key recovery information. Various levels of validation may be performed depending on the key recovery techniques used by the sending and receiving parties, including:

- Verification of the presence of the KRB,
- Validation of the integrity of the KRB,
- Authentication of the source and validation of the integrity of the KRB ~~[WILL THIS BE THE CASE? INFO MAY NEED TO BE ADDED]~~, and
- Verification that the KRI can be used to recover the data key.

The KRB is independent of the encryption algorithm used to protect the confidentiality of the data, and independent of the communication or storage protocol used to carry the encrypted data.

B.2 KRB Fields

The KRB should include the following fields of information:
[need a list, not alliance specific, only top level, …]

~~– The KRB version number,~~

~~– The KRB length – beginning at the version number and ending at the last word/byte of the Integrity Field,~~

~~– Object Identifier (OID) for the key recovery technique used to generate the KRI field.~~
~~– Encrypted Data Sensitivity (EDS) Field Type:~~

[12] Note that the KRB is not itself KRI - the KRB *contains* KRI plus other information, including an integrity checking value.

~~Type = 0: NONE (no EDS field is specified)~~

~~Encrypted Data Sensitivity (EDS) Field Length:~~

~~Number of {words/bytes} in the EDS field.~~

~~Encrypted Data Sensitivity (EDS) Field– the sensitivity of the data recoverable by this KRB [THIS NEED WAS IDENTIFIED IN THE BUSINESS REQUIREMENTS PAPER PRODUCED BY THE KEY RECOVERY ALLIANCE – SEE SCENARIO 13, 2ND COLUMN, 4TH ITEM] .~~

~~KRI Field length – in {words,bytes}.~~

~~KRI Field (KRIF) – the KRI as specified by the indicated key recovery technique using the format employed by that technique,~~

~~Encrypted Data Locator (EDL) Field Type – identifies the method used to generate the EDL Field. Defined methods include:~~

~~type = 0: NONE (no EDL field was calculated)~~

~~Encrypted Data Locator (EDL) Field Length:~~

~~Number of {words/bytes} in the EDL field.~~

~~Encrypted Data Locator (EDL) Field:~~

~~The value of the Encrypted Data Locator Field. This is reserved for possible future use in locating the encrypted data that may be recovered using this KRB.~~

~~Integrity Field Type:~~

~~Identifies the method used to generate the Integrity Field. Defined methods include:~~

~~type = 0: NONE (no integrity field was calculated)~~
~~type = 1: SEMANTIC (no integrity field was calculated)~~
~~type = 2: PROTOCOL (no integrity field was calculated)~~
~~type = 3: CONF-HMAC-SHA-1-96 (integrity field calculated using HMAC and SHA-1 and the confidentiality key associated with the KRF - described in RFC 2104 and draft-ietf-ipsec-hmac-sha196-00.txt)~~

~~type = 4: CONF-HMAC-MD5-96 (integrity field calculated using HMAC and MD5 and the confidentiality key associated with the KRF~~

~~type = 5: INTEG-HMAC-SHA-1-96 (integrity field calculated using HMAC and SHA-1 and the integrity key associated with the session - described in RFC 2104 and draft-ietf-ipsec-hmac-sha196-00.txt)~~

~~type = 6: INTEG-HMAC-MD5-96 (integrity field calculated using HMAC and MD5 and the integrity key associated with the session - described in RFC 2104 and draft-ietf-ipsec-hmac-md5-96-00.txt)~~

~~type = 7: SIGNATURE-PKCS7 (integrity field calculated as a PKCS #7 envelope with ContentType = "signed data" - described in the PKCS #7 specification. The data content that is carried within the PKCS#7 envelope is the hash of the KRF. The hash algorithm used is the same one that is specified within the PKCS#7 Content.~~

~~Integrity Field Length:~~

~~Number of {words/bytes} in the Integrity Field. The Integrity Field Length must be consistent with the Integrity Field Type:~~

~~Integrity Field Type~~	~~Integrity Field Length~~
~~0~~	~~0~~
~~1~~	~~0~~
~~2~~	~~0~~
~~3~~	~~5~~
~~4~~	~~4~~
~~5~~	~~5~~
~~6~~	~~4~~
~~7~~	~~Varies~~

~~Integrity Field Value:~~

~~The value of the Integrity Field that is calculated over all fields of the KRB except for the Integrity Field Value itself.~~

~~For Integrity Field Types 0 through 2, the Integrity Field value does not exist. For Integrity Field Types 3 and 5, the Integrity Field Value is a 20 byte hash of the KRF using HMAC and SHA-1. For Integrity Field Types 4 and 6, it is a 16 byte hash of the KRF using HMAC and MD5.~~

~~For Integrity Field Type 7, the Integrity Field Value is a PKCS#7 envelope [SEE ENVELOPE STRUCTURE BELOW] whose content is a hash of the relevant fields of the KRB using the digestAlgorithmIdentifier specified within the PKCS#7 Content.~~

~~[NOTE: THE FOLLOWING MAY NEED TO BE REMOVED OR EXTENSIVELY REVISED BASED ON THE THE FIPS VALIDATION REQUIREMENTS.]~~

~~Further Notes on the Integrity Field:~~

~~Certain key recovery products do not require any verification of the KRIF to be done at the receiving side. These products should use Integrity Field Type "NONE", indicating that KRIF verification is unnecessary.~~

~~Certain other products use technique-specific validation methods for the KRIF since these may be potentially stronger than the KRIF integrity checking techniques that are supported by the KRB. Products of this class should construct KRBs with Integrity Field Type "SEMANTIC", implying that the KRIF should be validated semantically using the technique-specific algorithm. A major drawback of using semantic validation techniques is that interoperability between products using dissimilar key recovery techniques may not be supportable.~~

~~Some key recovery products are based on secure communication protocols which provide integrity protection for the KRB when it is carried as an integral part of the secure association. This class of products should use Integrity Field Type "PROTOCOL", implying that the KRIF need not be checked for integrity since the carrier protocol provides integrity protection for the entire KRB.~~

~~Finally, there are a class of key recovery products which require KRIF validation by the receiver who cannot rely on the carrier protocol to provide integrity protection to the KRB, and require interoperability between heterogeneous key recovery systems. This class of products should use the supported integrity checking mechanisms of the KRB by using Integrity Field Types 3 to 7. The Integrity Field should contain the value corresponding to the specified type.~~

~~Certain products may like to use keyed-hash based integrity checks for the KRB. These products will generate KRBs with Integrity Field Types 3 to 6. The keyed-hash Integrity Field Types are defined for systems that use a single key for confidentiality and integrity protection, as well as systems using separate confidentiality and integrity keys. Types 3 and 4 use the confidentiality key associated with the session in generating the HMAC value, while types 5 and 6 use the integrity key associated with a session for the HMAC. A careful analysis of the cryptographic system is required when the same key does double duty as the encryption key and the HMAC key for the key recovery block.~~

~~Certain products may like to use digital signature techniques to validate the integrity of the KRB. These products will generate KRBs with Integrity Field Type 7, which denotes that the Integrity Field Value is a PKCS#7 envelope that carries a digital signature over the relevant fields of the KRB. The PKCS#7 format was chosen as a vehicle for carrying the signature value since it allows the pertinent certificates (needed for signature verification) to be conveniently packaged in a well-known format. It may be noted that the Content within the PKCS#7 envelope is a hash of the relevant fields of the KRB. Thus, the actual signature carried within the PKCS#7 envelope will be calculated on the hash of the KRB, rather than the KRB itself.~~

~~It may be noted that a product that generates a KRB specifies the Integrity Field Type based on its assumptions about its operating environment and its policy related to KRIF verification. Similarly, the types of KRBs that may be accepted by a receiver product are based on the receiver's assumptions about its operating environment and its policy related to KRIF verification. This proposal in no way mandates that a receiver product accept a KRB with all possible integrity Field Types; it leaves the usage and acceptability of specific Integrity Field Types to the discretion of the sending and receiving products.~~

~~The KRB format can also be used very conveniently to identify the KRIF carried within it. Certain vendors may like to use the KRB format for KRIF identification purposes only, but may not want to incur the overhead of generating and verifying the integrity field. It is recommended that these vendors use Integrity Field Type "NONE".~~

~~The KRB integrity field is a "robust" mechanism for verifying the integrity of the enclosed KRIF. The integrity field is not susceptible to typical man-in-the-middle attacks (MITM). Modification of the Integrity Field Type is not useful to an attacker, since the communicating peers have a security association that demands specific Integrity Field Types. Substitution of the KRIF and the corresponding Integrity Field value (for types 3 to 6) does not succeed, since the MITM does not have the session key necessary to generate a valid Integrity Field value for the bogus KRIF that was substituted.~~

~~B.3 KRB Format~~

~~[NEED TO DECIDE HOW DEEPLY TO SPECIFY THE KRB. SHOULD PROBABLY SPECIFY ENOUGH THAT ANY PROTOCOL USING THE KRB INFORMATION WOULD USE THE SAME STRUCTURE, ALLOWING DEVELOPERS TO DESIGN TO THE SAME FORMAT.]~~

~~31~~	~~23~~	~~15~~	~~7~~
~~Version~~		~~KRB Length~~	
~~Key Recovery Technique OID~~			
~~EDS Field Type~~		~~EDS Field Length~~	
~~Encrypted Data Sensitivity (EDS)~~			

~~Reserved~~	~~KRIF Length~~
~~Key Recovery Information Field (KRIF)~~	
~~EDL Field Type~~	~~EDL Field Length~~
~~Encrypted Data Locator (EDL) Field~~	
~~Integrity Field Type~~	~~Integrity field length~~
~~Integrity Field Value~~	

~~B.4 Implementation Guidance~~

~~Vendors that are compliant with the common KRB format, would design their products so that their KRIF (in its proprietary format) is placed within the common KRB defined above. The appropriate information should be provided in the other fields of the KRB. When a compliant product receives a KRB with an integrity field, the product can validate the KRIF embedded within the KRB, either by using the KRB integrity field, or the technique-specific validation algorithm (if known).~~

~~----------------~~

~~[THE PRESENCE OF THE PKCS#7 STRUCTURE IS FOR INFORAMTION PURPOSES ONLY.]~~

~~PKCS #7 Structure~~

~~PKCS #7 contains the following fields:~~

- ~~Version~~
- ~~Digest Algorithm ID~~
- ~~Content~~
- ~~Certificates (opt.)~~
- ~~CRLs (Opt.)~~
- ~~Signer Info~~

    ~~version,~~
    ~~issuer & serial ID,~~
    ~~digest algorithm ID,~~
    ~~authenticated attribute (opt.),~~
    ~~encrypted digest,~~
    ~~unauthenticated attributes (opt.)~~

~~Note: There may be multiple Signer Info fields~~

Appendix C: Certificate Extensions

[steve to re-word]

C.1 ~~Introduction~~KRA Certificate

In order to facilitate the recovery of a key in a Public Key Infrastructure (PKI), the following extended key usage OID will be registered. This key usage OID can be employed to identify a public key of a KRA that will be used to encrypt KRI. ~~the appropriate certificates should be extended to include key recovery information. Modifications may include:~~ The extended key usage should be marked critical, to ensure appropriate use of the corresponding public key.

{joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) keypurpose(2) krakey(1)}

- ~~The encryption certificate for a KRA should include:~~

~~(1) a key usage bit which indicates that the public key is to be used for key recovery purposes.~~

~~ASN.1 for this has two alternatives:~~

~~a. Recommend adding a bit to the X.509 key usage extension: keyRecoveryAgent (9) and mark the key usage extension critical, or~~

~~b. Register the following object as the key purpose object identifier and mark it critical. This seems to be the more appropriate approach rather than expecting X.509 to add a bit to the key usage extension.~~

~~{joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) keypurpose(2) krakey(1)}~~

~~(2) an identification of the key recovery technique(s) with which the public key may be used.~~

~~It is not clear that this extension is required at all. Santosh suggests not having this extension at all. Note that if this extension is used, the first extension is not required since this extension implies that the public key belongs to the KRA. Thus, it is recommended that if this extension is used,~~

~~the first extension (extended key usage) should be dropped. The following is the syntax for this private, critical[13] extension to the certificate:~~

~~**kRTechnique EXTENSION ::= {**~~

~~**SYNTAX KRTechnique**~~

~~**IDENTIFIED BY id-extensions-kRTechnique }**~~

~~**KRTechnique ::= SEQUENCE {**~~
~~**technique technique.&id,**~~
~~**parameters OPTIONAL }**~~

~~*-- technique is an object identifier. The parameters syntax is registered when the technique OID is registered*~~

C.2 Subscriber Certificate

~~—~~A certificate for a subscriber to ~~the entity using~~ a key recovery service should include: [need more intro, describe rationale for each extension, …]

(1) an indication that the entity has a key recovery capability, This is done by using the following private, non-critical extension

```
keyRecoveryCapable EXTENSION ::= {
     SYNTAX          SubjectKeyIdentifier
     IDENTIFIED BY   id-extensions-KeyRecoveryCapable }
KeyRecoveryCapable ::= BOOLEAN DEFAULT FALSE
```

(2) identify the KRA(s), This is done using the following private, non-critical extension. ~~Please n~~Note that if this extension is included, the first extension (key recovery capable) ~~is not required~~need not be present.

```
kR EXTENSION ::= {
     SYNTAX          KR
     IDENTIFIED BY   id-extensions-KR }
```

[13] The extension must be critical since only those who understand it in the key recovery context should use this public key.

```
KR ::= SEQUENCE SIZE (1...MAX) OF KRS

KRS ::= SEQUENCE {

      technique        KRTechnique

      SEQUENCE SIZE (1...MAX) OF AGENT }

kRTechnique EXTENSION ::= {

     SYNTAX                    KRTechnique

     IDENTIFIED BY             id-extensions-kRTechnique }

KRTechnique       ::=      SEQUENCE {
     technique        technique.&id,
     parameters       OPTIONAL }


  --  technique is an object identifier. The parameters syntax is
  registered when the technique OID is registered


AGENT ::= SEQUENCE {

     agentName         general~~directory~~Name

     agentkey          KeyIdentifier – OPTIONAL

     agentpol          KRAPolicy – OPTIONAL}


KRAPolicy ::= OBJECT IDENTIFIER
```

(3) indicate the KRA certificate(s) containing the appropriate KRA public key(s). Please note that this is in the KR extension.

(4) identify the key recovery technique(s) supported by the entity. Please note that this is in the extension for the key recovery (item 2 above)

(5) include any key recovery technique information required. Please note that this is in optional parameters extension of the key recovery technique.

[reword and move into C.2 intro text]

CSOR REGISTERED TECHNICAL OBJECTS

Prefix for CSOR-unique technical objects: {2.16.840.1.101.3}

{joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3)}
[provide some text to put this in context, e.g., note that this arc is maintained by NIST.]

-- Technical Object Identifiers

-- Types of information security objects

id-slabel	ID ::= {id-csor 1}
id-pki	ID ::= {id-csor 2}
id-arpa	ID ::= {id-csor 3}

-- Certificate Policies
-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) certpolicies(1)}

-- Key Purpose
-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) keypurpose(2)}

id-kRAKey	ID ::= {id-keypurpose 1}

-- Extensions
-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) extensions(3)}

id-kRTechnique	ID ::= {id-extensions 1}
id-kRecoveryCapable	ID ::= {id-extensions 2}
id-kR	ID ::= {id-extensions 3}

-- Key Recovery Schemes
-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) keyrecoveryschemes(4)}

-- Key Recovery Policy
-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) krapol(5)}

~~Appendix D: Interoperability Examples~~

~~D.1 Introduction~~

~~D.2 S/MIME~~

~~The Secure MIME (S/MIME) protocol provides encryption for Internet electronic mail that uses the MIME encoding format. S/MIME defines two security wrappers: one for digital signatures and one for encryption. To encrypt and sign a message, both wrappers are applied. Both of these wrappers build on the formats defined in PKCS#7 version 1.5. For encryption, the EnvelopedData wrapper is used. The EnvelopedData wrapper requires RSA key management, and the RSA public keys must be carried in certificates.~~

~~S/MIME does not include a location that can be used to carry a key recovery field. However, the key recovery center could be a recipient on every message, even if the message is not delivered to the key recovery center. In this way, the key recovery center private key can be used to recover the message plaintext content.~~

~~Key recovery may also be done as part of certificate management. This technique only works if the originator is a recipient of the message. That is, a RecipientInfo field for the originator must be included to ensure that the key used to encrypt the message content is available to the key recovery center who holds a copy of the originator's RSA private key.~~

~~D.3 MSP~~

~~The Message Security Protocol (MSP) provides encryption for Internet and X.400 electronic mail. MSP is used in the Defense Message System, and MSP is specified in SDN.701. Like S/MIME, MSP supports both digital signatures and encryption; however, MSP defines one wrapper to provide both services. MSP is algorithm independent.~~

~~MSP includes two locations that could be used to carry a key recovery field: the token and the extensions. To carry a key recovery field in the token, a separate object identifier for a new key management technique must be assigned. This approach would destroy interoperability with existing implementations. To carry a key recovery field in the extensions, a non-critical extension is added to the end of the message. MSP does not encrypt the extensions; therefore a key recovery field carried in an extension would be accessible.~~

~~Alternatively, the key recovery center could be a recipient on every message, even if the message is not delivered to the key recovery center. In this way, the key recovery center private key can be used to recover the message plaintext content.~~

~~Key recovery may also be done as part of certificate management. MSP includes a token for the originator. If the mail transfer system is unable to deliver the MSP protected message and returns the message to the originator as part of non-delivery notification, this token allows the originator to decrypt the message to determine which one was returned. If the key recovery center holds a copy of the originator's private key, then the key recovery center can also use the originator token to decrypt the message content.~~

~~D.4 PEM~~

~~The Privacy Enhanced Mail (PEM) protocol provides encryption for Internet electronic mail. PEM defines one encapsulation mechanism for digital signatures and encryption. PEM is defined in Internet RFCs 1421 through 1424. Two forms of key management are supported for encryption: RSA key management using certificates, and out-of-band distribution of symmetric key encryption keys.~~

~~PEM includes one location that could be used to carry a key recovery field: the Key-Info header line. This header line is used for both forms of key management. To carry a key recovery field in the Key-Info line, a separate Date Encryption Key protection algorithm identifier must be assigned. This approach would destroy interoperability with existing implementations.~~

~~Key recovery may also be done as part of certificate management. RFC 1421 recommends that a Key-Info header line be included for the originator as well as each recipient. This technique only works if the originator Key-Info header line is included. That is, a Key-Info header line for the originator must be included to ensure that the key used to encrypt the message content is available to the key recovery center who holds a copy of the originator's RSA private key. RFC 1424 specifies the certificate management for PEM, and a single RSA key is used for key management and digital signature. Thus, this form of key recovery permits a malicious key recovery center to masquerade as the originator by generating signed PEM messages. These unauthorized messages could also be encrypted.~~

~~D.5 ISAKMP~~

Appendix E: Key Recovery Techniques

[fix KRI validation mentions, since that is a configurable feature]
[revisit subsection titles, move to be new Appendix A]
[move the following definition into the body of the standard, in section 2.]

Cryptographic end systems that satisfy this key recovery standard use key recovery techniques which may be broadly categorized into two types, ~~KRI~~ key encapsulation and key escrow. The ~~KRI~~ key encapsulation technique associates key recovery information with the encrypted data in a manner which allows the KRA to recover the data key. The key escrow technique makes the cryptographic end system's key, usually a long term key such as a public/private key pair, directly accessible by a KRA. This appendix provides an overview of these two techniques.

E.1 ~~KRI~~ Key Encapsulation

Figure 9 illustrates the interaction of two cryptographic end systems that share or communicate encrypted data using a ~~KRI~~ key encapsulation technique for key recovery. To make the data key recoverable, the KRI Generation Function within the Cryptographic End System labeled A (hereinafter referred to as System A) first generates (or acquires) and encapsulates KRI corresponding to the data key. Then, the KRI is provided to the KRI Delivery Function.

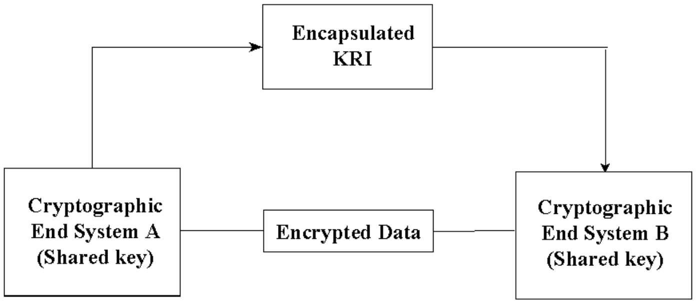

Encapsulated KRI
Cryptographic End System A (Shared key)
Encrypted Data
Cryptographic End System B (Shared key)

Figure 9: KRI Encapsulation Technique

Cryptographic End System labeled B (hereinafter referred to as System) may receive the KRI as well as the encrypted data and key exchange information. The KRI received by System B may be processed to a KRI Validation Function. Whether and what type of validation is performed is dependent on the structure and content of the KRI, the key recovery technique used, and the validation policy of the receiving cryptographic end system.

This method works equally well where System A and System B are actually the same system, as would be the case in a storage application.

2.9.2 Key Escrow

Figure 10 illustrates [fix figure to make 2 KRAs] the interaction of two cryptographic end systems that share or communicate encrypted data using a key escrow technique for key recovery. For each cryptographic end system, keys, key parts or key related information to be recovered are delivered to and stored at the KRA. In this technique, a third party or a cryptographic end system acts as a KRI Provider, generating and delivering KRI to KRA(s).

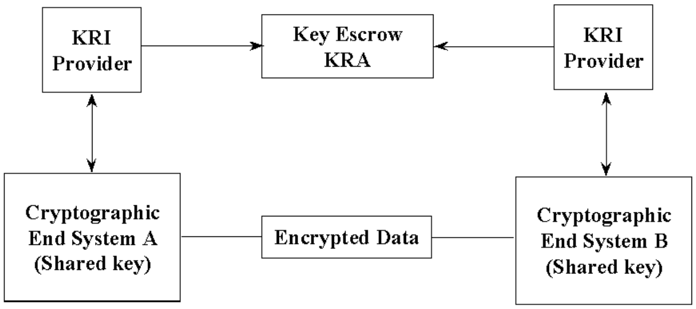

KRI Provider
Key Escrow KRA
KRI Provider
Cryptographic End System A (Shared key)
Encrypted Data
Cryptographic End System B (Shared key)

Figure 10: Key Escrow Technique Technique

In an environment where System A is encrypting data and sending it to System B, a key escrow scheme allows System A to make the target key recoverable without the addition of encapsulated KRI. System A can determine that System B is using an acceptable key escrow technique for key recovery by acquiring this information from some source (e.g., a certificate) using its KRI Validation Function. In this case, System A's normal performance of the key exchange/negotiation protocol may be sufficient to make the target key recoverable.

If required to do so, System B may verify recoverability by verifying that its own public key has been escrowed. This allows the normal performance of the key exchange/negotiation protocol to make the data key recoverable.

E.3 Interactions Between Systems Using Different Key Recovery Techniques

Cryptographic end systems that interact with systems using different key recovery techniques may still provide for key recovery. Furthermore, cryptographic end systems may provide for key recovery even when communicating with systems with no key recovery capability.

E.3.1 Interactions Between ~~KRI~~ Key Encapsulation and Key Escrow Techniques

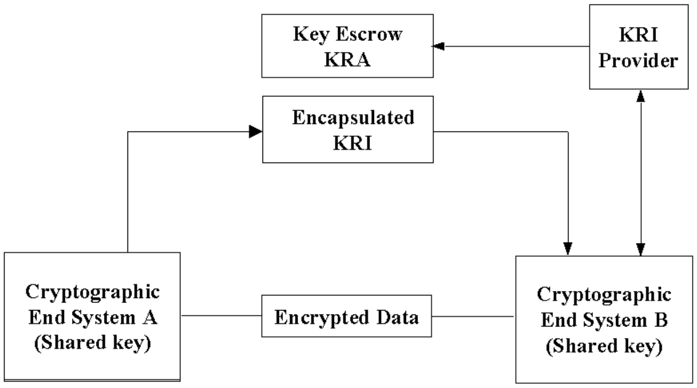

Key Escrow KRA
KRI Provider
Encapsulated KRI
Cryptographic End System A (Shared key)
Encrypted Data
Cryptographic End System B (Shared key)

Figure 11: KRI Encapsulation-based System Interaction with Key Escrow-based System

In Figure 11 System A uses a ~~KRI~~ key encapsulation technique to provide for key recovery, whereas System B uses a key escrow technique. System A may be able to use its KRI Validation Function to determine that System B uses key escrow. System A can create encapsulated KRI using its KRI Generation Function and provide it to its KRI Delivery Function. System B's KRI Provider must independently provide KRI to System B's KRA prior to any possible recovery of System B's key. In this case, System B does not need to validate the encapsulated KRI since System B's key has been escrowed, though may optionally choose to do so.

In Figure 12, System A uses a key escrow technique to provide for key recovery, whereas System B uses a ~~KRI~~ key encapsulation technique. For System A to provide for key recovery, encapsulated information must be provided (e.g., by encrypting a copy of the data key for System A and placing it in a recipient list or in a key recovery block) using the KRI Generation and Delivery Functio performance of the key exchange mec functions.

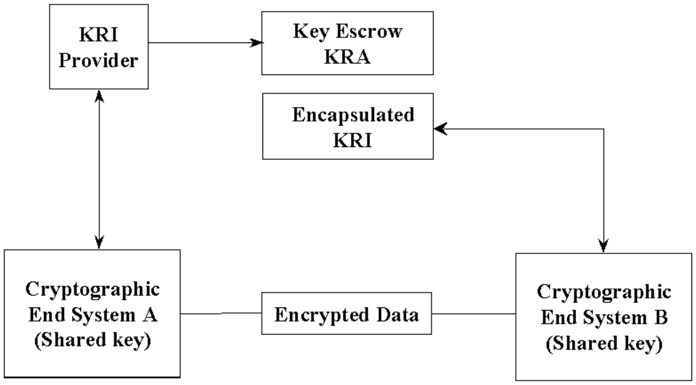

KRI Provider
Key Escrow KRA
Encapsulated KRI
Cryptographic End System A (Shared key)
Encrypted Data
Cryptographic End System B (Shared key)

Figure 12: Key Escrow-based System Interaction with KRI Encapsulation-based System

System B may be able to use its KRI Validate Function to determine the type of key recovery employed by System A and check for the presence of encapsulated KRI. If System B must either validate or provide for the data key's recoverability, System B may be able to generate and deliver encapsulated KRI in accordance with its key recovery technique.

E.3.2 Interactions Between KRI Encapsulation and Systems with No Key Recovery

In Figure 9, if System A uses ~~KRI~~ key encapsulation and System B has no key recovery capability, System A can provide encapsulated KRI even though System B cannot attempt to verify its recoverability. The encapsulated KRI received from System A must not cause interoperability problems with System B, however (see Section 2.7).

If the roles are reversed and System B initiates a communication, System A's KRI Validation Function will detect that System B has not provided suitable KRI. If System A must either validate or provide for the data key's recoverability, System A may be able to generate and deliver encapsulated KRI.

E.3.3 Interaction Between Key Escrow and Systems with No Key Recovery

In Figure 12, if System A uses Key Escrow and System B has no key recovery capability, System A can ensure the recoverability of the communication only if encapsulated information is created by its own KRI Generation and Delivery Functions (e.g., by encrypting a copy of the data key for System A and placing it in a recipient list or in a key recovery block). System A must ensure that System B will be able to ignore the presence of the KRI in order to permit interoperability.

If the roles are reversed and System B sends encrypted data to System A, System A can recover if the data key is recoverable using System A’s escrowed key.