REQUIREMENTS FOR KEY RECOVERY PRODUCTS

(ADVISORY COMMITTEE WORKING DRAFT)

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

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,
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:

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 can be traced uniquely to the entity.
Assurance	The degree of confidence that a product correctly implements the security functions. In the context of this FIPS, three levels of assurance are specified, representing increasing degrees of confidence.
Auditable 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 verify the claimed identity of an entity, e.g., a password, PIN, biometric, or response to a challenge.
Authentication Mechanism	A technique used to verify the claimed identity of an entity, e.g., user ID and password, token, biometric, or challenge-response.
?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 operations.
Common Criteria (CC)	An international standard for security in information security products. (See Cross Index.)
Confidentiality	The property that information is not made available or disclosed to an unauthorized user, process, or object.
Configurable	Aproperty whereby a feature of a product ~~that~~ may or may not be enabled.
Configuration Item	An item (e.g., documents, software, hardware) 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 (non-recoverable)	An encryption product for which the output is not recoverable.
Cryptographic End System (Recoverable)	See Section 2.6.
Cryptographic Module (cryptomodule)	A set of hardware, firmware, software, or some combination thereof 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 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 a Key Recovery Requestor function.
Decryption	A process for transforming ciphertext into plaintext, using a cryptographic algorithm and a key.
Encryption	A process for transforming plaintext into ciphertext through the use of a cryptographic algorithm and a key.
Evidence of Origin	A proof of the origin of information that cannot be (successfully) repudiated by the originator, e.g., a message digitally signed by the originator.
Evidence of Receipt	A proof of the receipt of information that cannot be (successfully) repudiated by the recipient , e.g., a digitally signed receipt issued by the recipient of the message.
FIPS 140-1	FIPS 140-1 specifies security functionality and assurance requirements for cryptomodules. See Cross Index.
FIPS Compliant	Meeting all requirements of a specified level of a FIPS.
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.
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 in natural language, e.g., English.
Integrity	The property that data has not been modified in an unauthorized and

	undetected manner.
Interoperability	The ability of products or systems to communicate with one another.
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 with the encrypted data.
Key Escrow	A method of key recovery in which the secret or private keys, key parts, or key related information to be recovered is stored by one or more Key Recovery Agents.
Key Recovery Agent (KRA) Function	A key recovery system function that performs a recovery service in response to an authorized request.
Key Recovery Product	A product that performs one or more key recovery system functions.
Key Recovery Information (KRI)	All or part of the information that is required for the recovery of a key. The KRI does not include a plaintext key.
Key Recovery Block (KRB)	A data structure that serves as a container for a single key recovery scheme-specific KRI and associates the KRI with a set of standard fields in a predefined format.
Key Recovery Policy	A policy 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)	A function in a key recovery system used to request keys.

Function	
Key Recovery System (KRS)	This consists of the KRI generation , the KRI management , and the key recovery . It includes software, hardware, procedures, and infrastructure.
KRI Delivery Function	A key recovery system function that assembles and formats key recovery information (KRI) and makes it available for recovery and validation.
KRI Generation Function	A key recovery system function that generates all or part of the key recovery information (KRI).
KRI Validation Function	A key recovery system function that authenticates or verifies key recovery information.
Least Abstract Representation	The most concrete representation of an implementation (e.g., source code).
Presentation of Evidence	Providing information necessary to carry out the assurance activity.
Private Key	A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and not made public.
Public Key	A cryptographic key used with a public key cryptographic algorithm, uniquely associated with an entity, and which may be made public.
Realtime Communication	Communication in which data transmission between the sender and receiver is intended to take place in near real time, for contemporaneous

	communication. Virtual terminal emulation and the world wide web are examples of real time communication. Contrast with staged delivery communication.
Recovery Registration Information (RRI?)	Information provided to a KRA in support of (later) key recovery.
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.).
Secret Key	A cryptographic key used with a secret key [symmetric] cryptographic algorithm, known by one or more entities, and not be made public.
Security Domain	A set of security-related services, mechanisms, and policies.
Security Policy	A set of rules and procedures regulating the use of information, including its processing, storage, distribution, and presentation.
Self Recovery	Key recovery effected by a subscriber (or subscriber organization?), in contrast to key recovery performed by an unaffiliated third party, e.g., as a service.
Staged Delivery Communications	Communication in which data transmission between the sender and receiver is stored at one or more intermediate points, e.g., to facilitate communication when not both the sender and the receiver are simulaneously available. Electronic mail is the best known example of staged delivery communication. Contrast with real time communication.
Standard Communication	Any communication protocol adopted by a generally recognized standards organization. For this standard, the phrase “standard communication

Protocol	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).
Subject	An active entity, generally in the form of a person, process, or device that causes information to flow among objects (passive entities that contain or receive information) or changes the system state.
Target Key	The cryptographic key recovered by a Key Recovery System.
Target Key Information (TKI)	Information provided by a KRA in response to an authorized key recovery request.
Trusted Path	A mechanism by which a person or process can communicate directly with a key recovery system function and which can be activated only by the person, process, or the function.
Trusted Time Stamp	A date and time that is reliable, accurate, and is affixed in such a way as to preclude undetected modification.

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 Information Technology (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, that 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. Three levels of compliance are defined: Level 0, Level 1 and Level 2. Level 0 provides a low level of security for a Key Recovery Requestor Function providing self-recovery, whereas Levels 1 and 2 provide medium and high levels of security functionality for other functions and key recovery scenarios. 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[1], and represent a profile of that security assurance evaluation criteria for use in this context. Three levels of (increasing) security 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 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. This appendix 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 C describes the concept of a Key Recovery Block (KRB), a data structure that 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 the requirements specified in Section 2.

Appendix D defines 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. Many KRS designs make use of public key certificates. The extensions defined here provide a standard means of representing certain data that is 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 mandates neither the use of X.509 certificates nor these extensions.

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

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 encryption key (DEK) that can be used to decrypt the data, or
- a key that can be used to decrypt the encrypted DEK.

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), 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).

Two types of key recovery techniques have been addressed in this standard, key encapsulation and key escrow. The key encapsulation technique associates key recovery information with the encrypted data in a manner which allows the KRA to recover the DEK. 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.

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.

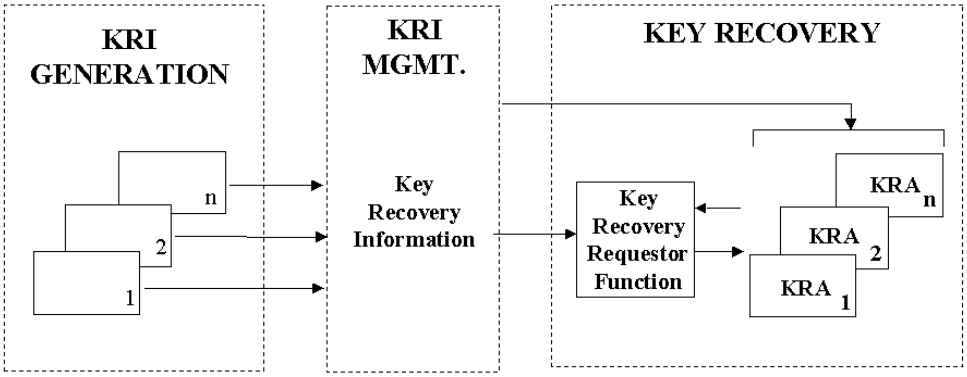

Figure 1: General Model for Key Recovery Systems

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

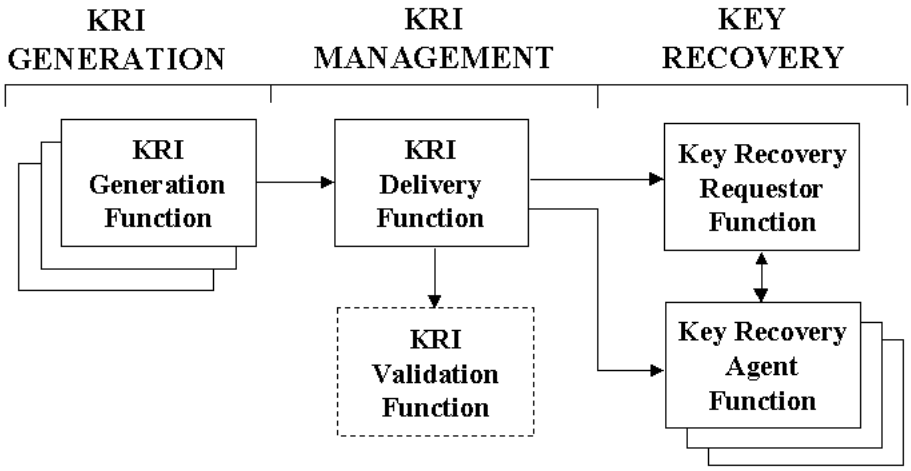

Figure 2: The Five Functions of a Key Recovery System

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

- Realtime communication sessions
- Staged delivery 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 B 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.

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 operate in the product and how they 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.

(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. 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. 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. 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. A KRI-generating entity could, for example, be the sender or receiver of a communication, a Certification Authority (CA), a Key Distribution Center, 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 DEK/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 DEK which was used to encrypt the ciphertext data. See Appendix B 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) in order to allow key recovery. For example, for key encapsulation systems , 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, 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. 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. 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. 8) The KRI Generation Function shall provide the generated KRI to the KRI Delivery Function.

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

KRI for a given DEK/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 DEK), 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. 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 Function directly, or depositing the KRI in one or more locations known to and accessible by the Key Recovery Requestor Function.

(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. 14) The KRI Validation Function shall be configurable.

[THE ISSUE OF CONFIGURABILITY NEEDS TO BE REVISITED. SPECIFICALLY, A PRODUCT NEED NOT BE CONFIGURABLE IF THERE ARE NO "NON-COMPLIANT" FEATURES - ONLY IF NON-COMPLIANT FEATURES ARE PRESENT AND NEED TO BE SWITCHED OFF/ON FOR THE PRODUCT TO OPERATE IN A FIPS COMPLIANT MODE. THIS SHOULD JUST REQUIRE A CAREFUL WORDING.]

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 KRI validation is enabled (i.e., turned on), it may prevent interoperation between two cryptographic end systems.

(Req. 15) 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 DEK 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, the 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 (KRR) Function

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

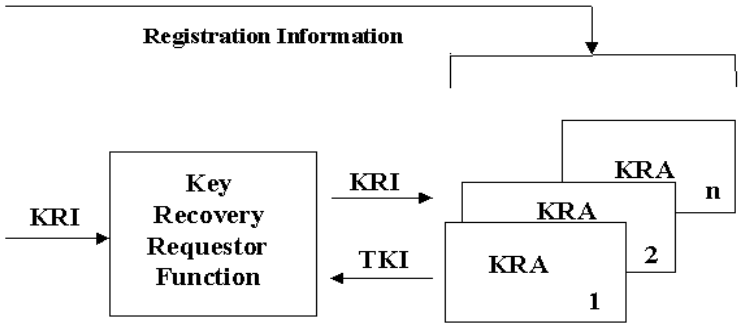

Figure 3: Key Recovery Functions

(Req. 16) For given KRI, the 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 KRR Function. The KRR Function interacts with one or more KRAs to obtain either a target key, multiple key parts, or key related information which will allow the reconstruction of 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 components which are then combined to reconstruct the target key.

(Req. 17) Encrypted data transmitted by the KRR Function shall be recoverable.

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

2.5 Key Recovery Agent Function

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

(Req. 18) The KRA shall store keys, key components or any other information required to satisfy the recovery of a target key .

(Req. 19) All of the data needed to operate the KRA, and all cryptomodules employed by the KRA, must be securely replicable, in support of availability.

The 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; the 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. 20) 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 performed by a KRA consists of processing all or part of the KRI provided to the KRA by the KRR Function, and returning an output value to the KRR Function. 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. 21) Encrypted data transmitted by the KRA Function shall be recoverable.

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. 22) 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 DEKs 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 B). 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, the application 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

This standard establishes interoperability requirements for cryptographic end systems. No interoperability requirements are imposed on communication between a cryptographic end system and a Key Recovery Agent (KRA), among KRAs, or between a KRR and a KRA. In the latter cases, 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.

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 standardcommunication 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. 23) 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. The inclusion of such a configurable

feature does not disqualify a system relative to (Req. 23) above. 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 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 systems capable of key recovery, and between such systems and those that are not capable of key recovery. 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 be configurable to 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 D for a proposed X.509 certificate extension.

If a key encapsulation scheme is employed, KRI may 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[2], PGP[3], S/MIME[4], or X.411[5] an additional recipient, representing a KRA, could be added to the per-recipient token list to provide key recovery on a per message basis.

One may also support key encapsulation in an interoperable fashion by transmitting KRI via a parallel communication path that does not impinge on the target communication protocol, i.e., the protocol for which key recovery is being effected. For example, in the Internet environment one might transmit KRI in the payload of ICMP Echo messages, exchanged between the parties whose communication is intended to be key recoverable. Because the processing of ICMP Echo messages is a standard feature of Internet protocol implementations and does not require parsing of the payload, this approach to key recovery meets the interoperability requirements established in this Standard.

---

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

[3] REFERENCE NEEDED

[4] Secure Multipurpose Internet Mail Extension

[5] 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[6] incorporates an optional field in the Pick-SA-Attrs exchange to carry KRI. In ISAKMP[7], one party could transmit a (yet to be defined) NOTIFY message with a payload containing per-session KRI. A compliant ISAKMP implementation would 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 within the target protocol, 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. 24) 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.

[6] IEEE 802.10c/D6, Standard for Interoperable LAN Security-Part C: Key Management.
[7] 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. Examples include, but are not limited to: a product on which no unevaluated software or firmware can be loaded, a product on which no other software or firmware can be loaded, and a product built to run on a general purpose operating system (e.g., UNIX, Windows NT, etc.). The requirements for the various key recovery system functions have been defined so that any product architecture can be evaluated.

A product architecture may imply that some of the requirements are satisfied, since the threats the requirements are supposed to mitigate, do not arise in that architecture. 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. In such situations, the product is considered compliant with this standard with respect to the requirements in question.

Some of the requirements may be satisfied by the underlying general purpose system software, such as the operating system, and/or DBMS. For example, the underlying operating system may satisfy the identification and authentication, and audit requirements.

[CHECK TO SEE IF INTRODUCTION OF SELF RECOVERY NOTION, E.G., LEVEL 0 KRR, INTERACTIONS BADLY WITH THE FOLLOWING KRA REQUIREMENTS.]

This section requires that the key recovery functions are implemented in products which conform to FIPS 140-1, levels 1,2,or 3.

- Level 1 specifies basic security requirements for a cryptomodule. 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.

- Level 2 improves upon the physical security of a Level 1 cryptomodule by (a) requiring tamper evident coatings or seals, or pick-resistant locks, (b) requiring role-based authentication and (c) allowing software cryptography in multi-user timeshared systems when used in conjunction with a C2[8] or equivalent operating system.

- Level 3 improves upon the Level 1 and 2 requirements for cryptomodules by (a) requiring tamper detection mechanisms, (b) requiring identity-based authentication, (c)

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

specifying stronger requirements for entering and outputting critical security parameters, and (d) 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.

3.1 Key Recovery Information Generation Function

3.1.1 Level 1 – Medium Security Key Recovery Information Generator

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

3.1.1.1 Cryptographic Functions

(Req. 25) All cryptographic modules shall be compliant with FIPS 140-1, Level 1 or higher.

3.1.1.2 Cryptographic Algorithms

(Req. 26) A KRI Generation 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

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. 27) Transmitted KRI must be protected via encryption. The strength of the algorithm used to protect the KRI 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.

[9] ANSI X9.44, Key Management Using Reversible Public Key Cryptography for the Financial Services Industry.

3.1.1.4 Integrity

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

(Req. 28) The KRI Generation Function shall generate an integrity value for the KRI.

(Req. 29) The KRI Generation Function shall associate the KRI with the encrypted data.

(Req. 30) 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 DEK meets all of the above three requirements. (Req. 28) is met since the keyed message digest provides integrity. (Req. 29) 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. 30) 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.

3.1.1.5 Identification and Authentication

(Req. 31) All cryptographic modules shall implement role-based authentication.

(Req. 32) The cryptographic module shall include a system administrator role.

3.1.1.6 Access Control

(Req. 33) The KRI Generation Function shall allow only a system administrator to configure this function.

(Req. 34) 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 3.1.1.5.

3.1.2 Level 2 – High Security Key Recovery Information Generator

3.1.2.1 Cryptographic Functions

(Req. 35) All cryptographic modules shall be compliant with FIPS 140-1, Level 2 or higher.

3.1.2.2 Cryptographic Algorithms

Same as Level 1.

3.1.2.3 Confidentiality

Same as Level 1.

3.1.2.4 Integrity

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

(Req. 36) 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.

3.1.2.5 Identification and Authentication

No additional requirements at this level.

3.1.2.6 Access Control

No additional requirements at this level.

3.2 Key Recovery Information Delivery Function

No Security requirements.

3. 3 Key Recovery Information Validation Function

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

3.3.1 Level 1 – Medium Security Key Recovery Information Validation Function

3.3.1.1 Cryptographic Functions

(Req. 37) All cryptographic modules shall be compliant with FIPS 140-1, Level 1 or higher.

3..3.1.2 Cryptographic Algorithms

(Req. 38) A KRI Validation Function which is submitted for evaluation shall be able to be configured to use only FIPS approved algorithms (where applicable).

3..3.1.3 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.

(Req. 39) Prior to decrypting the data, the KRI Validation Function (if enabled) shall verify the integrity value for the KRI .

(Req. 40) Prior to decrypting the data, the KRI validation Function (if enabled) shall verify the association of the KRI with the encrypted data.

(Req. 41) Prior to decrypting the data, the KRI Validation Function (if enabled) shall verify the integrity value for the association of the KRI to the encrypted data.

See Section 3.1.1.4 “Key Recovery Information Generation Function – Integrity” for an example of how the above integrity requirements can be satisfied.

3.3.2 Level 2 – High Security Key Recovery Information Validator

3.3.2.1 Cryptographic Functions

(Req. 42) All cryptographic modules shall be compliant with FIPS 140-1, Level 2 or higher.

3.3.2.2 Cryptographic Algorithms

Same as Level 1.

3.3.2.3 Integrity

(Req. 43) 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.

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

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

(c) 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.

3.4 Key Recovery Requestor Function

3.4.1 Level 0 - Low Security

For this function, a third, lower level of security requirements is defined. The primary motivation for this additional level is self-recovery.

3.4.1.1 Cryptographic Functions

(Req. 44) All cryptographic modules shall be compliant with FIPS 140-1, Level 1 or higher.

3.4.1.2 Cryptographic Algorithms

(Req. 45) A Key Recovery Requestor Function which is 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 KRR Function 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 this algorithm, e.g., use of RSA[10] for key encapsulation.

3.4.1.3 Confidentiality

There are no confidentiality requirements imposed at this level.

3.4.1.4 Integrity

(Req. 46) 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.

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

(Req. 47) 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.

3.4.1.5 Audit

There are no audit requirements imposed at this level.

3.4.1.6 Identification and Authentication (I&A)

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

3.4.1.7 Access Control

There are no requirements imposed at this level for access control.

3.4.1.8 Authentication of Received Transactions

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

3.4.1.9 Non-Repudiation

(Req. 48) The product shall provide time stamps for use in transactions with the KRA Function.

(Req. 49) The product shall generate evidence of origin for key recovery requests.

3.4.1.10 Protection of Trusted Security Functions

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

3.4.2 Level 1 – Medium Security

3.4.2.1 Cryptographic Functions

(Req. 50) All cryptographic modules shall be compliant with FIPS 140-1, Level 2 or higher.

3.4.2.2 Cryptographic Algorithms

Same requirements as Level 0.

3.4.2.3 Confidentiality

(Req. 51) The KRR Function shall protect both received and stored TKI against disclosure to unauthorized individuals.

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

(Req. 52) The KRR Function 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. 53) If a KRR Function 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. 54) 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.

3.4.2.4 Integrity

Same requirements as Level 0.

3.4.2.5 Audit

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

(Req. 55) The Key Recovery Requestor (KRR) Function shall cease operation if it is unable to effect audit operations.

(Req. 56) The product shall generate an alarm to a security administrative role if the size of the audit data in the audit trail exceeds a pre-defined limit.

(Req. 57) The product shall provide a security administrative role with the ability to manage the audit trail at any time during the operation of the product.

(Req. 58) Keys shall not be included in audit trails.

(Req. 59) 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 the 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 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. 60) It shall not be possible to disable the auditing of an event defined as "always audited."

(Req. 61) The following events shall always be audited:
(a) Requests, notifications, and other transactions generated by the product;
(b) Responses and other transactions received by the product, including key recovery responses; and
(c) Start-up and shutdown of the audit functions.

(Req. 62) 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 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 a security administrative role, the role, user identity and the addition/deletion action shall be included in the audit record.
(5) For all transactions, the entire transaction (excluding keys and TKI) shall be included in the audit record as sent or received.

(Req. 63) The product shall be able to generate a human understandable presentation of any audit data stored in the permanent audit trail.

(Req. 64) The audit trail shall not store the old or new authentication information (e.g., passwords)

(Req. 65) The product shall be able to associate each auditable event with the identity of the user that caused the event.

(Req. 66) The product shall provide a security administrative role with the ability to empty the audit trail.

(Req. 67) 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. 68) The product shall restrict access to the audit trail to a security administrative role.

3.4.2.6 Identification and Authentication

The requirements in this section are for the identification and authentication of KRR Function personnel (KRR 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 Function 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 a crypto officer is invoking a KRR cryptographic module function, authentication may be effected directly to the module and is exempt from all of the requirements of this section. In this case, the FIPS 140-1 level 2 module I&A requirements apply.

(Req. 69) The product shall provide functions for initializing and modifying KRR personnel authentication data.

(Req. 70) The product shall restrict the use of initialization and modification of the KRR personnel authentication data to a security administrative role.

(Req. 71) The product shall allow authorized KRR personnel to modify their own authentication data.

(Req. 72) The product shall protect authentication data that is stored in the product from unauthorized observation, modification, and destruction.

(Req. 73) 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. 74) The product shall be able to terminate the session establishment process after at most five unsuccessful authentication attempts.

(Req. 75) After the termination of the session establishment process, the product shall be able to disable the KRR personnel account until the account is enabled by a security administrative role.

(Req. 76) The product shall authenticate the claimed identity of an individual prior to performing any functions on behalf of that individual.

(Req. 77) 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.

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. 78) If the product makes use of a "trusted path" mechanism to meet the preceding I&A requirement, that trusted path between itself and the KRR personnel shall be logically distinct from other communication paths and shall provide an assured identification of its endpoints. KRR personnel shall have the ability to initiate communication via this trusted path.

3.4.2.7 Access Control

(Req. 79) The product shall verify the association of the response to an outstanding request.

(Req. 80) The product shall provide an ability to destroy TKI and target keys, e.g., by zeroizing.

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. 81) The product shall ensure that security features are always invoked and cannot be bypassed.

(Req. 82) The product shall maintain a security domain for its own execution that protects the product from interference and tampering by untrusted subjects.

(Req. 83) The product shall enforce separation between the security domains of subjects in the system.

(Req. 84) 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. 99)).

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 authorized requestors ,

- product cryptographic key management,
- actions on the audit log, audit profile management, and
- changes to the system configuration.

(Req. 85) The product shall be capable of distinguishing the set of KRR personnel authorized for administrative functions from all other personnel.

(Req. 86) The product shall allow only specifically authorized KRR personnel to assume a security administrative role.

(Req. 87) The product shall require an explicit request to be made in order for authorized KRR personnel to assume a security administrative role.

3.4.2.8 Authentication of Received Transactions

Same requirements as at Level 0.

3.4.2.9 Non-Repudiation

Same requirements as Level 0, except that the following requirement replaces (Req. 48).

(Req. 88) The product shall provide trusted time stamps for use in transactions with the KRA Function.

3.4.2.10 Protection of Trusted Security Functions

(Req. 89) Before establishing a session with an individual, the product shall display an advisory warning message regarding unauthorized use of the product.

(Req. 90) 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. 91) The product shall restrict the capability to modify the warning message to an authorized security administrative role.

(Req. 92) Upon successful session establishment, the product shall display the date, time, method, and location of the last successful session establishment for the individual establishing the session.

(Req. 93) 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. 94) The data specified above shall not be removed from the display device without intervention by the individual establishing the session.

3.4.3 Level 2 – High Security

3.4.3.1 Cryptographic Functions

(Req. 95) All cryptographic modules shall be compliant with FIPS 140-1, Level 3 or higher.

3.4.3.2 Cryptographic Algorithms

Same as Level 0.

3.4.3.3 Confidentiality

Same as Level 1.

3.4.3.4 Integrity

Same as Level 1.

3.4.3.5 Audit

Includes all the requirements of Level 1 and the following:

(Req. 96) 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**.

3.4.3.6 Identification and Authentication

Level 2 enhances security 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 a crypto officer is invoking a KRA cryptographic module function, authentication may be effected directly to the module and is exempt from the following requirement. In this case, the FIPS 140-1 level 3 module I&A requirements apply.)

All Level 1 requirements except that (Req. 77) is replaced by the following:

(Req. 97) The product shall support hardware token-based authentication. The token shall meet the requirements of FIPS 140-,1 Level 2 or higher.

3.4.3.7 Access Control

All Level 1 requirements apply as well as the following:

(Req. 98) Two or more individuals shall be required to request the TKI from the KRA Function.

(Req. 99) 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. 100) An individual in the system administrator role shall perform the following functions:

- **(a) the assignment/deletion of KRR personnel accounts,**
- **(b) the assignment/deletion of authorized KRR personnel to/from security administrative roles, and**
- **(c) the association of security-relevant administrative commands with security administrative roles.**

(Req. 101) 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. 102) The crypto officer shall manage the cryptographic keys.

(Req. 103) The audit administrator shall manage the audit log and audit profiles.

(Req. 104) The product shall associate each security-relevant administrative function with exactly one security administrative role.

(Req. 105) 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.

3.4.3.8 Authentication of Received Transactions

Same as Level 0.

3.4.3.9 Non Repudiation

Same as Level 1.

3.4.3.10 Protection of Trusted Security Functions

All Level 1 requirements apply as well as the following:

(Req. 106) The product shall provide the system operator role with the capability to demonstrate the correct operation of the security-relevant functions provided by the underlying abstract machine.

(Req. 107) The product shall preserve a secure state when abstract machine tests fail.

These two requirements ensure that the particular hardware system on which KRR software is operating is operating correctly(Req. 106) can be met by providing comprehensive integrity or diagnostic tests on the hardware. (Req. 107) can be met by terminating the KRR operations in case of hardware integrity or diagnostic test failure.

3.5 Key Recovery Agent Function Requirements

3.5.1 Level 1 – Medium Security

3.5.1.1 Cryptographic Functions

(Req. 108) All cryptographic modules shall be compliant with FIPS 140-1, Level 2 or higher.

3.5.1.2 Cryptographic Algorithms

(Req. 109) A KRA function submitted for evaluation shall be able to be configured to use only FIPS approved algorithms (where applicable).

See (Req. 26) for additional clarifying details.

3.5.1.3 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. 110) 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 (Req. 17).

(Req. 111) The KRA Function shall protect target key information transmitted - either electronically or physically communicated - against disclosure to unauthorized individuals.

(Req. 112) 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. 113) 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.

3.5.1.4 Integrity

(Req. 114) The product shall protect all stored KRI [and RRI ?]against modification.

(Req. 115) 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. 116) 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.

3.5.1.5 Audit

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

(Req. 117) The KRA shall cease operation if it is unable to effect audit operations.

(Req. 118) The product shall generate an alarm to a security administrative role if the size of the audit data in the audit trail exceeds a pre-defined limit.

(Req. 119) The product shall provide a security administrative role with the ability to manage the audit trail at any time during the operation of the product.

(Req. 120) Keys shall not be included in audit trails.

(Req. 121) 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 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. 122) It shall not be possible to disable the auditing of an event defined as "always audited."

(Req. 123) 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. 124) 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 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 a security administrative role, the role, user identity and the addition/deletion action shall be included in the audit record.
(5) For all transactions, the entire transaction (excluding keys and TKI) shall be included in the audit record as sent or received.

(Req. 125) The product shall be able to generate a human understandable presentation of any audit data.

(Req. 126) The audit trail shall not store old or new authentication information (e.g., passwords).

(Req. 127) The product shall be able to associate each auditable event with the identity of the user that caused the event.

(Req. 128) The product shall provide a security administrative role with the ability to empty the audit trail.

Note: emptying the audit trail means backup and delete.

(Req. 129) 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. 130) The product shall restrict access to the audit trail to a security administrative role.

3.5.1.6 Identification and Authentication (I&A)

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 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 requirements of this section. In this case, the FIPS 140-1 level 2 module I&A requirements apply.

(Req. 131) The product shall provide functions for initializing and modifying KRA personnel authentication data.

(Req. 132) The product shall restrict the use of initialization and modification of the KRA personnel authentication data to asecurity administrative role.

(Req. 133) The product shall allow authorized KRA personnel to modify their own authentication data.

(Req. 134) The product shall protect authentication data that is stored in the product from unauthorized observation, modification, and destruction.

(Req. 135) 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. 136) The product shall be able to terminate the session establishment process after at most five consecutive unsuccessful authentication attempts.

(Req. 137) After the termination of the session establishment process, the product shall be able to disable the user account until the account is enabled by a security administrative role .

(Req. 138) The product shall authenticate the claimed identity of an individual prior to performing any functions on the behalf of that individual.

(Req. 139) 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.

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. 140) If the product makes use of a "trusted path" mechanism to meet the preceding I&A requirement, that trusted path between the product and KRA personnel shall be logically distinct from other communication paths and shall provide an assured identification of its endpoints. KRA 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 or KRI generator. The requirements in this section address the security of electronic communication between the KRA and the KRR Function 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. 141) 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. 142) The product shall release target key information only to authorized requestors.

(Req. 143) 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.

(Req. 144) The product shall ensure that security features are always invoked and cannot be bypassed.

(Req. 145) The product shall maintain a security domain for its own execution that protects it from interference and tampering by untrusted subjects.

(Req. 146) The product shall enforce separation between the security domains of subjects in the system.

(Req. 147) 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. 168).

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 subscriberswhose 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. 148) The product shall be capable of distinguishing the set of KRA personnel authorized for administrative functions from all other personnel.

(Req. 149) The product shall allow only specifically authorized KRA personnel to assume a security administrative role.

(Req. 150) The product shall require an explicit request to be made in order for an authorized KRA operator to assume a security administrative role.

3.5.1.8 Authentication of Received Transactions

(Req. 151) The product shall verify the source of received transactions.

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

3.5.1.9 Non-Repudiation

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

(Req. 153) The product shall provide trusted time stamps for use in transactions with the KRR Function.

(Req. 154) The product shall generate evidence of origin for transmitted key recovery responses.

(Req. 155) If the product receives KRI [RRI?], the product shall generate evidence of receipt for that KRI.

(Req. 156) The product shall verify evidence of origin for key recovery requests and for KRI [RRI?] transactions.

3.5.1.10 Protection of Trusted Security Functions

(Req. 157) Before establishing a session with a KRA administrator, the product shall display an advisory warning message regarding unauthorized use of the product.

(Req. 158) 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) The product shall restrict the capability to modify the warning message to a security administrative role.

(Req. 160) 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. 161) 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. 162) The data specified above shall not be removed from the display device without intervention by the individual establishing the session.

3.5.2 Level 2 – High Security

3.5.2.1 Cryptographic Functions

(Req. 163) 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.

3.5.2.2 Cryptographic Algorithms

Same as Level 1.

3.5.2.3 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. 164) The system shall be designed for multiple KRAs. Two or more KRAs shall be required for a requestor to obtain the target key.

3.5.2.4 Integrity

Same as Level 1.

3.5.2.5 Audit

Level 2 adds a real time alarm to 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. 165) 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.**

3.5.2.6 Identification and Authentication (I&A)

Level 2 enhances security 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 a crypto officer is invoking a KRA cryptographic module function, authentication may be effected directly to the module and is exempt from the following requirement. In this case, the FIPS 140-1 level 3 module I&A requirements apply.)

All Level 1 requirements apply, except that (Req. 139) is replaced by the following:

(Req. 166) The product shall support a hardware token-based authentication. The token shall meet the requirements for FIPS 140-1, Level 2 or higher.

3.5.2.7 Access Control

Level 2 requires multi-party access controls for the release of KRI, and establishes roles and responsibilities for KRA 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. 167) 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. 168) 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. 169) 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 subscribers whose keys are held, and**
- **(d) the assignment/deletion of parties who may be provided the keys.**

(Req. 170) The system operator shall change the system configuration, execute abstract machine tests, change the advisory warning message, and operate the system.

(Req. 171) The crypto officer shall manage the cryptographic keys.

(Req. 172) The audit administrator shall manage the audit log and audit profiles.

(Req. 173) The product shall associate each security-relevant administrative function with exactly one security administrative role.

(Req. 174) 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.

3.5.2.10 Protection of Trusted Security Functions

All Level 1 requirements apply as well as the following:

(Req. 175) The product shall provide the system operator role with the capability to demonstrate the correct operation of the security-relevant functions provided by the underlying abstract machine.

(Req. 176) 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. 175) can be met by providing comprehensive integrity or diagnostic tests on the hardware. (Req. 176) can be met by terminating the KRA operations in case of hardware integrity or diagnostic test failure.

4 Assurance Requirements

Three Assurance Levels (ALs) are defined for this standard. Table 1 contains the classes, families, and components for the three ALs. Subsequent sections provide further detail. Section 4.8 contains the assurance requirements that are excluded from this standard.

(Req. 177) The KRA Function 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. 178) 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. 179) 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.

(Req. 180) A Theory of Compliance document shall address each requirement of this standard. For each requirement, the document shall contain a rationale as to how the product meets the requirement or explain why the requirement is not applicable.

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 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 evaluator is expected to examine, analyze, and/or test (as applicable) the assurance evidence for accuracy.

Table 1: KRS Assurance Levels

Assurance Class	Assurance Family	AL A	AL B	AL C
Configuration Management	CM Capabilities		4.1.1.2	4.1.1.3
	CM Scope			4.1.2.3
Delivery and Operation	Delivery		4.2.1.2	4.2.1.3
	Installation, Generation and Start-up	4.2.2.1	4.2.2.2	4.2.2.3
Development	Functional Specification	4.3.1.1	4.3.1.2	4.3.1.3
	High-Level Design	4.3.2.1	4.3.2.2	4.3.2.3
	Implementation Representation			4.3.3.3
	Low-Level Design			4.3.4.3
	Representation Correspondence			4.3.5.3
Guidance Documents	Administrator Guidance	4.4.1.1	4.4.1.2	4.4.1.3
	User Guidance	4.4.2.1	4.4.2.2	4.4.2.3
Life Cycle Support	Flaw Remediation	4.5.1.1	4.5.1.2	4.5.1.3
Tests	Coverage	4.6.1.1	4.6.1.2	4.6.1.3
	Depth	4.6.2.1	4.6.2.2	4.6.2.3
	Functional Tests	4.6.3.1	4.6.3.2	4.6.3.3
	Independent Testing	4.6.4.1	4.6.4.2	4.6.4.3
Vulnerability Assessment	Vulnerability Analysis		4.7.1.2	4.7.1.3

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 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.

A clear identification of the product is required in order 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 maintained by the CM system.

4.1.1.1 Configuration Management Capabilities – Assurance Level A Requirements

There are no requirements for configuration management capabilities at level A.

4.1.1.2 Configuration Management Capabilities – Assurance Level B Requirements.

(Req. 181) The developer shall use a CM system.

(Req. 182) The developer shall provide CM documentation that contains the following information:

(a) A configuration list describing the configuration items, and
(b) The method used to uniquely identify the product configuration items.

4.1.1.3 Configuration Management Capabilities – Assurance Level C requirements

Same as Level B (Section 4.1.1.2).

4.1.2 Configuration Management 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 Configuration Management Scope – Assurance Level A Requirements

There are no requirements for the scope of configuration management at this level.

4.1.2.2 Configuration Management Scope – Assurance Level B Requirements

There are no requirements for the scope of configuration management at this level.

4.1.2.3 Configuration Management Scope – Assurance Level C Requirements

At this level, a problem tracking system is required.

(Req. 183) 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. 184) The CM documentation shall also describe how configuration items are tracked by the CM system.

4.2 Delivery and Operation

4.2.1 Delivery and Operation– 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 Delivery – Assurance Level A Requirements

There are no assurance requirements at this level.

4.2.1.2 Delivery – Assurance Level B Requirements

At this level, delivery procedures are required.

(Req. 185) The developer shall provide documentation about the procedures for the delivery of the product or parts of the product to the user. The documentation shall describe the procedures to be employed when distributing versions of the product to a user's site.

4.2.1.3 Delivery – Assurance Level C Requirements

In addition to delivery procedures, the detection of unauthorized or accidental modification to the product is required at this level.

In addition to the Level B requirements, the following additional requirements are imposed.

(Req. 186) The delivery procedures documentation shall state how the procedures are to be employed to detect modifications.

(Req. 187) The delivery procedures 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. 188) The delivery procedures 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.

4.2.2 Delivery and Operation - 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 Installation, Generation, and Start-up – Assurance Level A Requirements

(Req. 189) The developer shall provide a document containing procedures and steps to be used for the secure installation, generation, and start-up of the product.

4.2.2.2

4.2.2.2 Installation, Generation and Start-up – Assurance Level B Requirements

Same as Level A.

4.2.2.3 Installation, Generation and Start-up – Assurance Level C Requirements

Same as Level A.

4.3 Development

4.3.1 Development - 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 should also include any additional specific detail specified by the documentation notes in the related functional products. For example, the functional specification should 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 Functional Specification and Security Policy – Assurance Level A Requirements

At this level, informal functional specification is required.

(Req. 190) The developer shall provide a complete and internally consistent functional specification that includes the syntax and semantics of the external product interfaces..

4.3.1.2 Functional Specification - Assurance Level B Requirements

At this level, an informal security policy model is added. All of the requirements from the previous assurance level and the following:

.

(Req. 191) The developer shall provide an informal security policy model. The informal security policy model shall:

(a) Describe the rules and characteristics of all policies of the product that can be modeled,
(b) Include a rationale that demonstrates that the policies that are modeled are satisfied by the informal security policy model, and
(c) Justify that all policies that can be modeled are represented in the informal security policy model.

(Req. 192) The developer shall provide a demonstration of the correspondence between the informal security policy model and the functional specification. The demonstration of correspondence between the informal security policy model and the functional specification shall:

(a) Describe how the functional specification satisfies the informal security policy model, and
(b) Show that there are no security functions in the functional specification that conflict with the informal security policy model.

4.3.1.3 Functional Specification – Assurance Level C Requirements

Same as Level B.

4.3.2 Development - 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

(Req. 193) The high-level design shall 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 High-Level Design – Assurance Level A Requirements

(Req. 194) The developer shall provide the high-level design of the product. The high-level design shall:

- **(a) Describe the structure of the product in terms of modules,**
- **(b) Describe the security functionality provided by each module of the product,**
- **(c) Identify the interfaces of the modules of the product, and**
- **(d) Identify any underlying hardware, firmware, and/or software required by the product with a presentation of the functions provided by the underlying hardware, firmware, or software**

4.3.2.2 High-Level Design – Assurance Level B Requirements

All Level A requirements apply in addition to the following.

(Req. 195) The high-level design shall describe the separation of the product into security policy enforcing modules and other modules.

4.3.2.3 High Level Design – Assurance Level C Requirements

Same as Levels A and B.

4.3.3 Development - 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 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 Implementation Representation – Assurance Level A Rerquirements

There are no implementation representation requirements at this assurance level.

4.3.3.2 Implementation Representation – Assurance Level B Rerquirements

There are no implementation representation requirements at this assurance level.

4.3.3.3 Implementation Representation – Assurance Level B Rerquirements

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.

(Req. 196) The developer shall provide the implementation representations for the product. . The implementation representations shall unambiguously define the product to a level of detail such that that an executable version of the product can be generated without further design decisions.

4.3.4 Development - 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 should 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 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 Low-Level Design – Assurance Level A Requirements

There are no low-level design requirements at this level.

4.3.4.2 Low-Level Design – Assurance Level B Requirements

There are no low-level design requirements at this level.

4.3.4.3 Low-Level Design - Assurance Level C requirements

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

(Req. 197) The developer shall provide the low-level design of the product. . The low-level design shall:

- **(a) Describe the product in terms of modules,**
- **(b) Describe the purpose of each module,**
- **(c) Define the interrelationships between the modules in terms of the functionality provided and the dependencies on other modules,**
- **(d) Describe the implementation of all security policy enforcing functions,**
- **(e) Describe the interfaces of each module in terms of their syntax and semantics,**
- **(f) Provide a demonstration that the product is completely represented, and**
- **(g) Identify the interfaces of the modules of the product which are visible at the external interface of the product.**

4.3.5 Development - 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 Representation Correspondence – Assurance Level A Requirements

There are no representation assurance requirements at this level.

4.3.5.2 Representation Correspondence – Assurance Level B Requirements

There are no representation assurance requirements at this level.

4.3.5.3 Representation Correspondence – Assurance Level C Requirements

(Req. 198) 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. 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.

4.4 Guidance Documents

4.4.1 Guidance Documents - 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 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 Administrator Guidance – Assurance Level A Requirements

(Req. 199) The developer shall provide administrator guidance addressed to system administrative personnel. The guidance shall contain the following information:

- **(a) How to administer the product in a secure manner,**
- **(b) Warnings about functions and privileges that should be controlled in a secure processing environment,**
- **(c) Guidelines on the consistent and effective use of the security functions within the product,**
- **(d) A description of 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,**
- **(e) A description of all security parameters under the administrator's control,**

(f) A description of each type of security-relevant event relative to the administrative functions that needs to be performed, including changing the security characteristics of entities under the control of the product,
(g) Guidelines on how the security functions interact,
(h) Instructions regarding how to configure the product,
(i) A description of all configuration options that may be used during the secure installation of the product, and
(j) A description of installation, configuration and operating procedures in sufficient detail for the administration of security.

4.4.1.2 Administrator Guidance – Assurance Level B Requirements

Same as Level A.

4.4.1.3 Administrator Guidance – Assurance Level C Requirements

Same as Level A.

4.4.2 Guidance Documents - 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 requirements 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 User Guidance - Assurance Level A requirements

(Req. 200) The developer shall provide user guidance containing the following information:

- **(a) A description of the product and interfaces available to the user,**
- **(b) Guidelines on the use of security functions provided by the product,**
- **(c) Warnings about functions and privileges that should be controlled in a secure processing environment, and**
- **(d) A description of the interaction between user-visible security functions.**

4.4.2.2 User Guidance – Assureance Level B Requirements

Same as Level A.

4.4.2.3 User Guidance – Assureance Level C Requirements

Same as Level A.

4.5 Life Cycle Support

4.5.1 Life Cycle Support - 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 Basic Flaw Remediation - Assurance Level A requirements

(Req. 201) The developer shall document the flaw remediation procedures that are used to track all reported security flaws in each release of the product. These procedures shall require the following information:

(a) A description of the nature and effect of each security flaw,
(b) The status of corrections to flaws, and
(c) The method for providing flaw and correction information to the users.

4.5.1.2 Flaw Remediation – Assurance Level B Requirements

All of the requirements from the previous assurance level, and the following additional requirements:

(Req. 202) The developer shall establish a procedure for accepting, tracking, and acting upon user reports of security flaws and requests for corrections to those flaws. The procedures for processing reported security flaws shall ensure that any reported flaws are corrected, and the correction is issued to product users.

4.5.1.3 Flaw Remediation – Assurance Level C Requirements

Same as Level B.

4.6 Tests

4.6.1 Tests - Coverage

Objectives
This family addresses those aspects of testing that deal with the completeness of testing. That is, this family 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 functional testing that is specified.

4.6.1.1 –Test Coverage – Assurance Level A Requirements

Objectives
In this component, the objective is that testing should 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.

(Req. 203) The developer shall provide an analysis of the test coverage demonstrating that the tests identified in the test documentation cover the product.

4.6.1.2 Test Coverage – Assurance Level B Requirements

Same as Level A.

4.6.1.3 Test Coverage – Assurance Level C Requirements

Same as Level A.

4.6.2 Tests - –Depth of Testing

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 functional tests selected.

4.6.2.1 Depth of Testing – Assurance Level A Requirements

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.

(Req. 204) The developer shall provide the analysis of the depth of testing in order to 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.

4.6.2.2 Depth of Testing – Assurance Level B Requirements

Same as Level A.

4.6.2.3 Depth of Testing – Assurance Level B Requirements

Same as Level A.

4.6.3 Tests - 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 testing 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 should eliminate all security relevant flaws discovered during testing.

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

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

4.6.3.1 Functional Testing Assurance Level A Requirements

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

(Req. 205) The developer is required to perform testing and to provide test documentation.

(Req. 206) The developer shall test the product and document the results.

(Req. 207) The developer shall provide test documentation consisting of test plans, test procedure descriptions, and test results. The test plans shall identify the security functions to be tested and describe the goal of the tests to be performed. The test procedure descriptions shall identify the tests to be performed and describe the scenarios for testing each security function. The test results in the test documentation shall show the expected results of each test. The test results from the execution of the tests by the developer shall demonstrate that each security function operates as specified.

4.6.3.2 Functional Testing Assurance Level B Requirements

Same as Level A requirements.

4.6.3.3 Functional Testing Assurance Level C Requirements

Same as Level A requirements.

4.6.4 Tests - 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 should 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) will execute these government supplied tests in addition to the tests provided by the developer, and the tests developed by the evaluator.

4.6.4.1 Independent Testing – Assurance Level A Requirements

At this level, executing a sample of vendor tests is sufficient.

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., functional testing)

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).

(Req. 208) The developer is required to perform testing and to provide test documentation and test results. This is addressed by the functional testing family.

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

(Req. 209) The evaluator shall execute a sample of tests in the test documentation in order to verify the developer test results.

4.6.4.2 Independent Testing – Assurance Level B Requirements

Same as Level A.

4.6.4.3 Independent Testing – Assurance Level C Requirements

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, as well as the supporting documentation and information required to run tests. The need for documentation is supported by other assurance families (e.g., functional testing)

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 functional testing family.

Replace the Level A assurance requirement with the following.

(Req. 210) 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 - 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 Vulnerability Analysis – Assurance Level A Requirements

There are no vulnerability analysis requirements at this level.

4.7.1.2 Vulnerability Analysis – Assurance Level B Requirements

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.

(Req. 211) 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. 212) The developer shall document the disposition of identified vulnerabilities.

(Req. 213) The evidence shall show, for each vulnerability, that the vulnerability cannot be exploited in the intended environment for the product.

4.7.1.3 Vulnerability Analysis – Assurance Level C Requirements

Same as Level B.

4.8 Excluded Assurance Requirements

The development assurances related to modularity, layering, information hiding, etc. have been excluded for economic reasons.

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

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

Some Misuse Analysis is included by including vulnerability analysis for obvious flaws and known flaws.

Appendix A: Key Recovery Techniques

This appendix provides an overview of the key encapsulation and key escrow key recovery techniques.

A.1 Key Encapsulation

Figure 4 illustrates the interaction of two cryptographic end systems that share or communicate encrypted data using a key encapsulation technique for key recovery. To make the DEK 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 DEK. Then, the KRI is provided to the KRI Delivery Function.

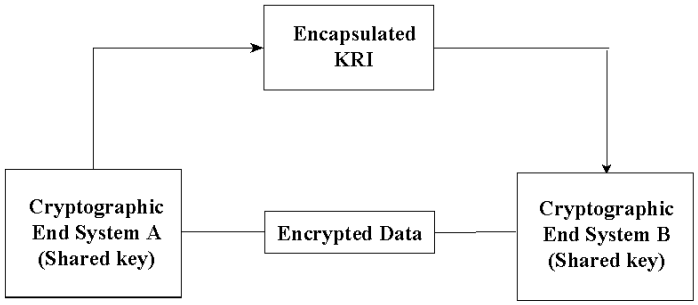

Figure 4: Key Encapsulation Technique

Cryptographic End System labeled B (hereinafter referred to as System B) may receive the KRI as well as the encrypted data and key exchange information. The KRI received by System B may be processed by a KRI Validation Function, if present and enabled. 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.

A.2 Key EscrowTechnique

Figure 5 illustrates the interaction of two cryptographic end systems that share or communicate encrypted data using a key escrow technique and different KRAs 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 the KRA(s).

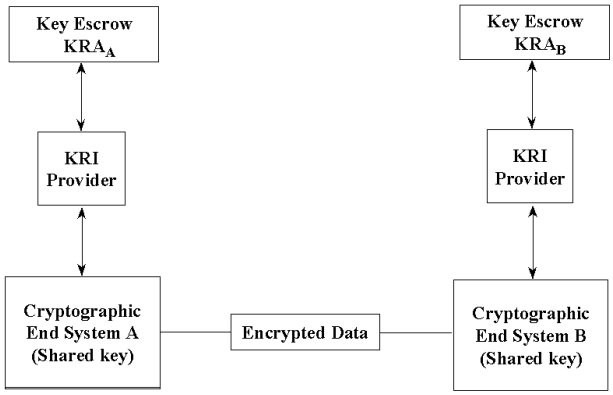

Figure 5: Key Escrow Technique Technique

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, if present and enabled. 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 DEK recoverable.

A.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.

A.3.1 Interactions Between Key Encapsulation and Key Escrow Techniques

In Figure 6, System A uses a 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 (if present and enabled) to determine that System B uses key escrow. System A can create encapsulated KRI using its KRI Generation Function and provide the encapsulated KRI 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 7, System A uses a key escrow technique to provide for key recovery, whereas System B uses a key encapsulation technique. For System A to provide for key recovery, encapsulated information must be provided (e.g., by encrypting a copy of the DEK for System A and placing the encrypted DEK in a recipient list or in a key recovery block) using the KRI Generation and Delivery Functions. Note that for some key exchange schemes, normal performance of the key exchange mechanism may provide for the KRI generation and delivery functions.

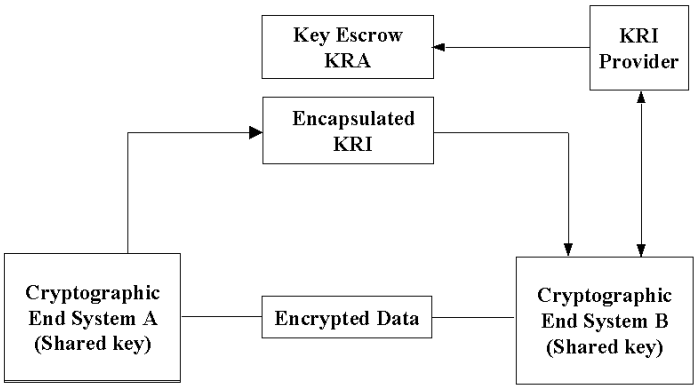

Figure 6: Key Encapsulation-based System Interaction with Key Escrow-based System

System B may be able to use its KRI Validate Function (if present and enabled) 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 DEK's recoverability, System B may be able to generate and deliver encapsulated KRI in accordance with its key recovery technique.

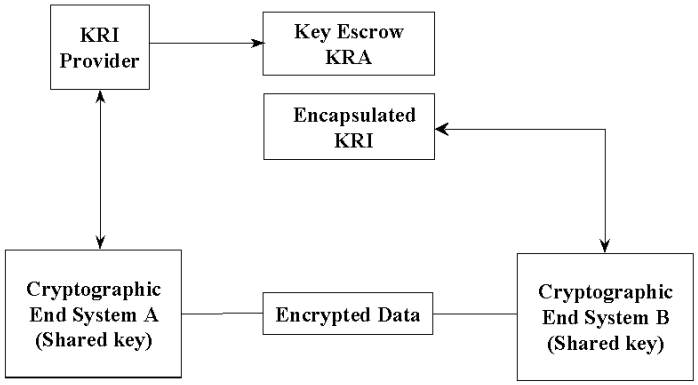

Figure 7: Key Escrow-based System Interaction with Key Encapsulation-based System

A.3.2 Interactions Between Key Encapsulation and Systems with No Key Recovery

In Figure 4, if System A uses key encapsulation and System B has no key recovery capability, System A can provide encapsulated KRI even though System B cannot 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 (if present and enabled) will detect that System B has not provided suitable KRI. If System A must either validate or provide for the DEK's recoverability, System A may be able to generate and deliver encapsulated KRI.

A.3.3 Interaction Between Key Escrow and Systems with No Key Recovery

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 DEK for System A and placing the encapsulated information 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 DEK is recoverable using System A's escrowed key.

Appendix B: Examples

B.1 Key Recovery Function Distribution

The functions of a KRS may be integrated into products in a variety of configurations in order to accommodate different user environments.

In Figure 8, 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.

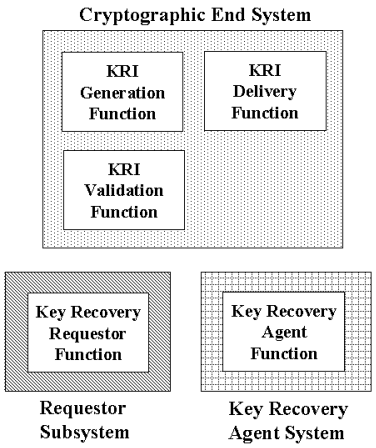

Figure 8

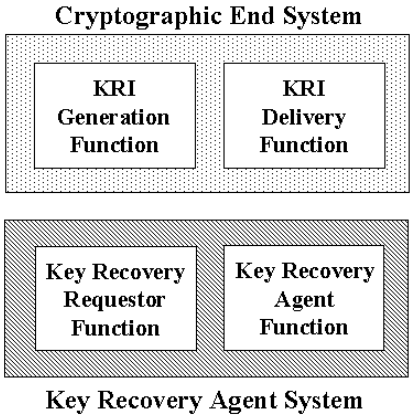

Figure 9

In Figure 9, 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.

In Figure 10, 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 11, 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.

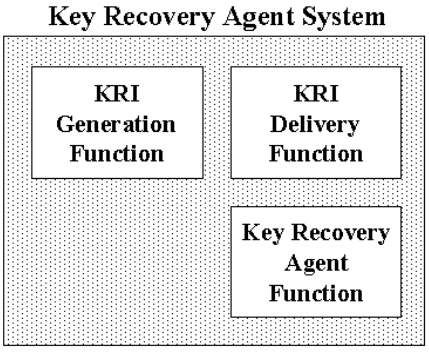

Figure 10

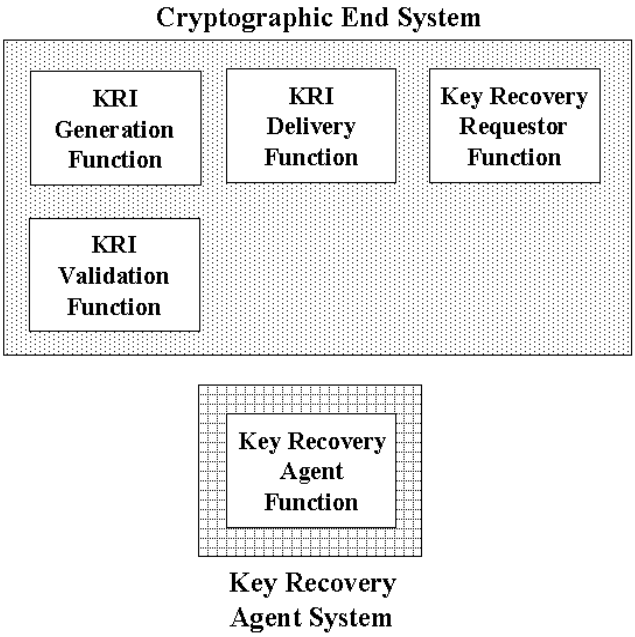

Figure 11

B.2 Multiple KRI Generation Functions

Figure 12 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.

B.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.

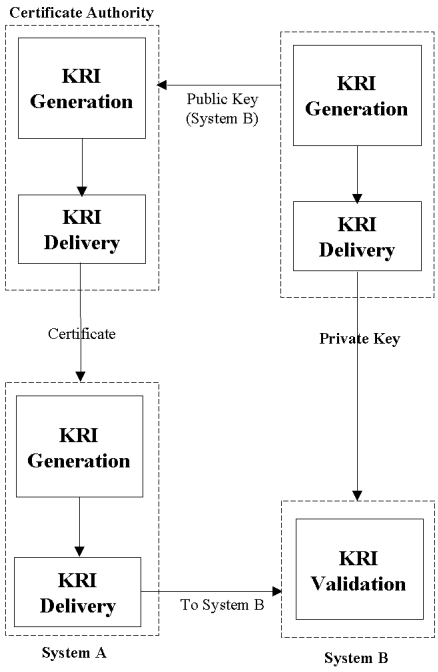

Figure 12

B.3.1 Realtime Communications

B.3.1.1 Between Two Encapsulation Techniques

In Figure 13, 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 DEK 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.

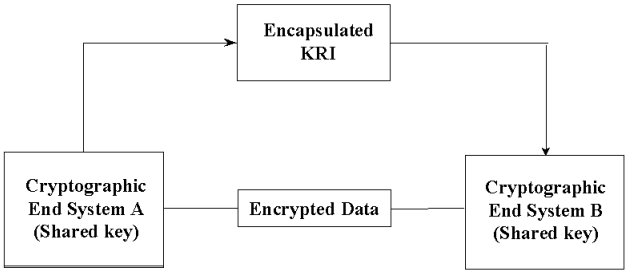

Figure 13

System A creates a DEK to be used for the communication session and encrypts the DEK 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 DEK (received as part

of the key exchange process), and uses the resulting plaintext DEK 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 DEK 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.

B.3.1.2 Between Encapsulated and Key Escrow Techniques

Figure 14 includes cryptographic end systems A and B that use key escrow and 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 DEK 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.

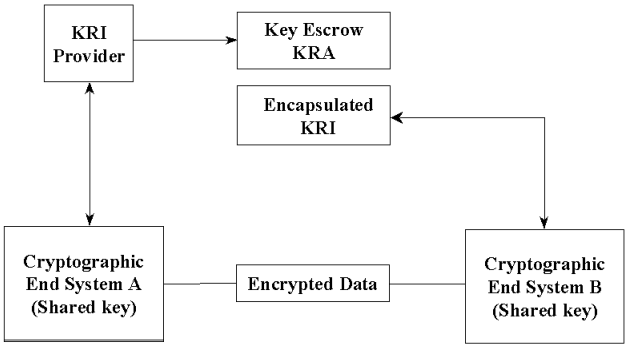

Figure 14

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 DEK, then encapsulates the DEK and other KRI in a KRB for itself and its KRA. The DEK 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.

B.3.2 Staged Delivery Communications

B.3.2.1 Between Two Key Escrow Key Recovery Schemes

In Figure 15, 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.

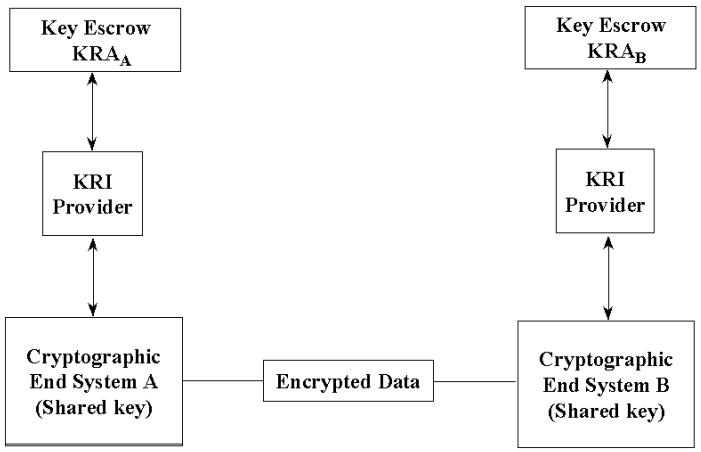

Figure 15

System B generates a DEK and encrypts the key using the encryption public key of the receiver (System A) 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 DEK using its own encryption public key, and places it in a KRB. System B then encrypts the message with the DEK, and sends the encrypted message and System A's copy of the encrypted DEK to System A. The encrypted message and the KRB are archived.

System A decrypts the DEK received via the key transport mechanism and decrypts the received message using that key.

B.3.2.2 Between an Encapsulated Scheme and an End User System with No Key Recovery Capability

In Figure 16 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 DEK 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.

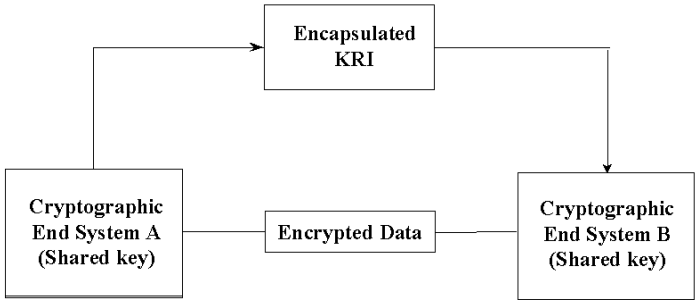

Figure 16

System A creates a DEK, 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 DEK 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.

B.3.3 Data Storage

B.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 16, cryptographic 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 DEK and uses the encryption public key of each system on the access list to encrypt a copy of the DEK for that system (including itself). System A also encrypts the DEK using the encryption public key of the organization's KRA. The DEK 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 DEK, and uses the decrypted DEK to decrypt the file.

B.4.1 Realtime Session

System A, a commercial telephone with a key recoverable encryption capability, is regularly used by a government employee whose job deals with high value contract actions. The key encrypting key used by the telephone to encrypt each session key (data encryption key) has been escrowed with the agency's KRA. The employee has come under suspicion for passing contract sensitive information to favored contractors in return for gratuities. The Inspector General's office has begun a serious investigation of this employee's activities. A request has been made to the appropriate law enforcement agency to conduct an in-house wire tap of the employee's telephone. The appropriate interception equipment has been set up at the agency's switchboard. Each encrypted call is recorded, and the KRI is parsed from the transmission. The KRI is forwarded to the agency's KRA for recovery of the key encrypting key needed to obtain the data encryption key. With the data encryption key and the recorded call, the clear content of the call can be recovered and reviewed for incriminating information.

B.4.2 Staged Delivery Communications

In scenario B.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 DEK for each message, and then uses the DEK to decrypt the associated message.

B.4.3 Data Storage

In scenario B.3.3.1, System A could create a private file (i.e., no one else is on the access list, so the DEK is not encrypted for anyone else). At some later time, the user needs to retrieve the file, but has lost access to the decryption key. The DEK 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.

Appendix C: Key Recovery Block

C.1 Overview

When different key recovery products that employ 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 key encapsulation, a common structure -- a Key Recovery Block (KRB) -- may be required. The KRB serves as a container[11] 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 , and
- Verification that the KRI can be used to recover the DEK.

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.

C.2 KRB Information

The KRB should include the following information:

- Identifier of the key recovery technique used to create the KRI,
- Indication of the sensitivity of the encrypted data[12],
- KRI created by the specified key recovery technique,
- Pointer to the encrypted data[13], and

---

[12] See "Business Requirements for Key Recovery", scenario 13 (col. 2, item 5), developed by the Key Recovery Alliance, 18 December 1997.

[13] When an explicit pointer to the encrypted data (e.g., the KRB and encrypted data are not attached or part of the same message).

- Integity value

Appendix D: Certificate Extensions

This appendix defines one certificate attribute value, for use in a certificate issued to a KRA, and one certificate extension, for use in a certificate issued to a subscriber whose private key has been escrowed with one or more KRAs.

D.1 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 by NIST. This key usage OID can be employed to identify a public key of a KRA that will be used to encrypt KRI. The extended key usage extension should be marked critical in order to ensure the appropriate use of the corresponding public key. (Because this certificate would not normally be used in conjunction with a standard protocol that is being targeted for key recovery, the critical marking does not violate the interoperability requirements established in this standard.

{joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) keypurpose(2) krakey(1)}

D.2 Subscriber Certificate

A certificate for a subscriber employing key recovery should include one of the following two extensions:

(1) The first extension provides an indication that the entity is employing a key recovery capability, This extension is private and non-critical. The value of this extension is a boolean.

```
keyRecoveryCapable EXTENSION ::= {
    SYNTAX          SubjectKeyIdentifier
    IDENTIFIED BY   id-extensions-KeyRecoveryCapable }
KeyRecoveryCapable ::= BOOLEAN DEFAULT FALSE
```

(2) The second extension identifies the key recovery technique(s) employed by the subscriber and, for each technique, identifies the KRA(s) that can be contacted to effect key recovery, relative to the technique. For each KRA, the extension optionally includes a key identifier (to specify the KRA's public key) and a KRA policy identifier. This extension is private and non-critical.

Note that if this extension is included, the first extension (**keyRecoveryCapable**) need not be present.

```
kR EXTENSION ::= {
    SYNTAX          KR
    IDENTIFIED BY   id-extensions-KR }
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       generalName
    agentkey        KeyIdentifier – OPTIONAL
    agentpol        KRAPolicy – OPTIONAL}

KRAPolicy ::= OBJECT IDENTIFIER
```

CSOR REGISTERED TECHNICAL OBJECTS

Prefix for CSOR-unique technical objects: {2.16.840.1.101.3}

The key recovery related objects will be registered under the NIST object registry. The following is the OID arc for NIST:

{joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3)}

-- Technical Object Identifiers

-- Types of information security objects

The following OIDs have already been registered.

id-slabel	ID ::= {id-csor 1}
id-pki	ID ::= {id-csor 2}
id-arpa	ID ::= {id-csor 3}

-- Certificate Policies

The certificate policy OID is as follows and has already been registered.

-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) certpolicies(1)}

-- Key Purpose

The following is a new recommended OID for the extended key usage purpose. First, a key purpose OID is registered under the PKI portion of the arc. The, a specific OID for the KRA encryption public key is defined.

-- {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

The following new OID is defined for key recovery related private certificate extensions. This is followed by OIDs for various key recovery related private 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

The following new OID is defined to accommodate various key recovery techniques (schemes). Since no scheme is registered yet, no OIDs for schemes are defined. As NIST registers key recovery schemes, they will be assigned OIDs under this arc.

-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) keyrecoveryschemes(4)}

-- Key Recovery Policy

The following new OID is defined to accommodate various key recovery policies. Since no key recovery policy is registered yet, no OIDs for policies are defined. As NIST registers key recovery policies, they will be assigned OIDs under this arc.

-- {joint-iso-ccitt(2) country(16) us(840) organization(1) gov(101) csor(3) pki(2) krapol(5)}