To: IEEE P1619.3 Task Group
From: P1619.3 Architecture Subcommittee, Members:

Walt Hubis, LSI (Chair)
Jon Holdman, Sun
Larry Hofer, Emulex
Luther Martin, Voltage Security
Glen Jaquette, IBM
Matt Ball, Independent
Robert Lockhart, nCipher
Landon Noll, Cisco
Robert Sussland, NetApp
Kevin Marks, Dell
Subhash Sankuratripati, Decru
Gaurav Agarwal, NetApp
Kevin Butt, IBM
Robert Griffin, RSA/EMC
Robert Snively, Brocade
Mike Witkowski, CipherMax (Past Chair)
Chris Williams, HP
Subbu Muddappa, Server Engines

Walt Hubis 3/7/08 8:14 AM
Deleted: Mike Witkowski, CipherMax (Chair)

Walt Hubis 3/7/08 8:15 AM
Deleted: Landon Noll, Cisco

Walt Hubis 3/7/08 8:17 AM
Deleted: Decru

Walt Hubis 3/7/08 8:17 AM
Deleted: Decru
Christopher Kusek, Network Appliance

Date: March 13, 2008
Purpose: Proposed changes against P1619.3/D1 to rewrite the contents and title of clause 4. This document proposes to change the title of clause 4 from "General Overview" to "General Concepts and Models". In addition, the contents of clause 4 will be rewritten with the goal to provide introductory concepts and models necessary to understand the remaining detailed information provided in the remaining clauses in the standard being developed by the other subcommittees in the P1619.3 Task Group.

Matthew Ball 3/10/08 1:27 PM
Deleted: March 7, 2008

Matthew Ball 3/10/08 1:27 PM
Deleted: March 7, 2008

Introduction

The P1619.3 Architecture Subcommittee has been working on a proposal to create a rewritten clause 4, entitled "General Concepts and Models" to incorporate a new set of models to help frame the material presented in the remainder of the standard. In addition, this section will be rewritten to provide a clearer introduction to the detailed clauses being developed by the other subcommittees in the P1619.3 Task Group.

Changes to P1619.3/D2

Matthew Ball 3/10/08 2:01 PM
Deleted: 1

The changes required by this proposal to draft P1619.3/D2 are:

Matthew Ball 3/10/08 2:01 PM
Deleted: 1

1) Rewrite of Clause 4, entitled "General Concepts and Models".
2) Addition of references to Normative References clause.
3) Addition and deletion of definitions, acronyms, and abbreviations for to the Definitions, Acronyms, and Abbreviations clause (Clause 3).
4) Addition of an informative annex that specifies the differences between the key management models proposed in the model clause of P1619.3 and the NIST SP 800-57 Part 1 standard and the ISO/IEC 11770-1 standard.
5) Move namespace subclause 4.4 out to Clause 5.

Matthew Ball 3/10/08 2:03 PM
Deleted: /ISO

Matthew Ball 3/10/08 2:01 PM
Formatted: Bullets and Numbering

2. Normative References

ASN X9.24 – Retail Financial Services Symmetric Key Management – Part 1: Using Symmetric Techniques.
ASN X9.24 – Retail Financial Services Symmetric Key Management – Part 2: Using Asymmetric Techniques for the Distribution of Symmetric Keys
[Matt: Move these references to the bibliography, or mention them by name in the main text]

3. Definitions, acronyms, and abbreviations

The following definitions in P1619.3/D1 must be removed:

- Client

Any replicate definitions, acronyms, and abbreviations in the following subclauses indicate that they are to replace the corresponding definitions, acronyms, and abbreviations specified in P1619.3/D1. Any new, unique definitions, acronyms, and abbreviations should be added to the draft.

3.1 Definitions

[Need to alphabetize this list eventually.]

3.1.1 Encryption Entity: A software and/or hardware entity that is capable of accepting cryptographic keys and using them to encrypt and decrypt data that is written and read, respectively, to persistent data storage media. Encryption entities are composed of a Cryptographic Unit and a Key Management Client.

3.1.2 Cryptographic Unit: An entity capable of using cryptographic algorithms to encrypt data being written to and decrypt data read from storage media. A Cryptographic Unit lies in both the data plane (where encryption and decryption actually take place) and the control plane (where data set information, cryptographic keys, and security parameters are transported between itself and the Key Management Client.

Matthew Ball 3/10/08 1:32 PM
Deleted: Cryptographic units implement the cryptographic algorithms specified in IEEE P1619, IEEE P1619.1, and IEEE P1619.2

3.1.3 Key Management Client: A component of an Encryption Entity that serves two primary functions: 1) communicates with Key Management Servers to acquire cryptographic keys, and 2) communicates with Cryptographic Units to load and activate these keys for their use in encrypting and decrypting data that is written and read, respectively, to data storage media.

3.1.4 Key Management Server: A software and/or hardware entity that is capable of generating, storing, protecting, and distributing cryptographic keys and other security related information necessary to support the operation of Encryption Entities. In addition, they may also support security services such has audit services, backup/archive services, security policy management, and access control services.

3.1.5 Key Management API: A collection of reference programming interfaces to the Key Management Software Library; one interface each for the C, C++, and Java language bindings.

3.1.6 Key Management Software Library: A software library developed by the P1619.3 Task Group that provides a reference implementation of a portion of the Key Management Client that generates Key Management Client-Server Operations. The interface to the library is the Key Management API. The library uses industry standard SSL/TLS and TCP interface calls for generating Key Management Client-Server Operations over the network. The use of this library is optional, but it can be used to facilitate the rapid development of compliant Key Management Clients by implementers.

3.1.7 Key Management Client-Server Operations: Defines the set of high level operations and their message and transport mechanisms that allow Key Management Clients to request key management services from Key Management Servers and, likewise, to allow Key Management Servers to respond to these key management service requests. These operations utilize a well defined set of key management objects for the communications of key management related information between the clients and servers.

3.1.8 Key Management Server-Server Operations: Defines the set of high level operations and their message and transport mechanisms that allow Key Management Servers to communicate with one another in order to facilitate the delivery of key management services that can not be accomplished with a single Key Management Server. Server-to-server operations facilitate advanced key management services such as hierarchical key distribution and redundancy and availability of key services for key management environments.

3.1.9 Data Plane: In the context of encryption of stored data, data plane operations occur then encryption users write plaintext data through the cryptographic unit where the data is encrypted and deposited as ciphertext to the storage media. Likewise, data plane operations also occur when encryption users read plaintext data through the cryptographic unit which decrypts ciphertext data retrieved from the storage media. Data plane operations tend to be high performance operations that are decoupled from the control plane where key management operations typically take place.

3.1.10 Control Plane: In the context encryption of stored data, control plane operations include those operations necessary to identify the encryption requirements of the stored data, generate cryptographic information cryptographic information necessary to properly configure the Cryptographic Units, retrieve this cryptographic information, and configure the Cryptographic Units with the acquired Cryptographic information. In addition, control plane operations include all operations necessary to protect and distribute this cryptographic information throughout a key management environment. Control plane operations tend to be lower performance operations that are decoupled from the data plane where the actual encryption of stored data takes place.

3.1.11 Encryption Users: Defines all users, servers, applications, and systems that require the stored data media used for their persistent storage requirements be protected by encryption.

3.1.12 Storage Media: Any device that is used to store and retrieve data over extended periods of time in a persistent manner. *[May need to check with the IEEE common terms and definitions for a better definition]*

3.1.13 Key Management User: One or more entities that are responsible for the configuration and management of the key management environment.

3.1.14 Key Management Objects: TBD

[add Encryption Key (or Cryptographic Key)] – Use definition from P1619.1

Policy: [See OO proposal for definition of Policy]

Matthew Ball 3/10/08 1:39 PM
Deleted: [Policy: need a definition]

Walt Hubis 3/7/08 8:14 AM
Formatted: Bullets and Numbering

3.2 Acronyms and abbreviations

CU	Cryptographic Unit
KM	Key Management
KM User	Key Management User
KM Client	Key Management Client

KM Server	Key Management Server
KMS	Key Management Service
KM API	Key Management Application Programming Interface
KMCS Ops	Key Management Client-Server Operations
KMSS Ops	Key Management Server-Server Operations
KM SW Lib	Key Management Software Library

4. General Concepts and Models

[Text for this intro to be drawn from D1 Word source file when available.]
TBD

4.1 Overview

[Text for this subclause to be drawn from D1 Word source file when available.]
TBD

4.1.1 Key Management Models

The following subclauses contains a number of abstract models that define the high level architecture for environments and systems that provide management for encryption keys and their related security attributes, parameters, and objects used for the protection of stored data. Industry and international standards define the operations and procedures for managing encryption keys and their related security material as "key management," and thus the objective of these subclauses is to present a series of key management models that provide an introduction to, and a context for, the detailed key management operations, objects, and protocols presented throughout this remainder of this standard.

Because of the complex nature of key management, several models have been developed to provide various viewpoints into the architecture of a key management environment and describe the entities and objects that make up these environments. The key management models presented in this standard are as follows:

- Key Management Architecture Model – the basic, high level model for key management environments that provides a foundation for the remaining models and detailed information developed in the later sections of this standard.
- Key Management Conceptual Model – provides the basic model for the internal and external organization of the primary components in a key management environment.
- Key Lifecycle Model – the basic state machine model that encompasses the entire set of standard key states and the transitions between these states as maintained by the key management systems that implement this standard.
- Key Management Sequence Models – a series of models that describe the high level interactions between the primary components in the Key Management Conceptual Model.
- Key Management Object Models
- Key Management Operation Models

Matthew Ball 3/10/08 1:52 PM
Formatted: Bullets and Numbering

4.2 Key Management Architecture Model

The protection of stored data using encryption requires the use of a key management environment where the encryption keys and other security related parameters can be adequately managed and protected. This standard defines a basic architecture for key management environment which includes the definitions and relationships between components within this environment.

Figure 1 illustrates the architecture model for the key management environment defined in this standard.

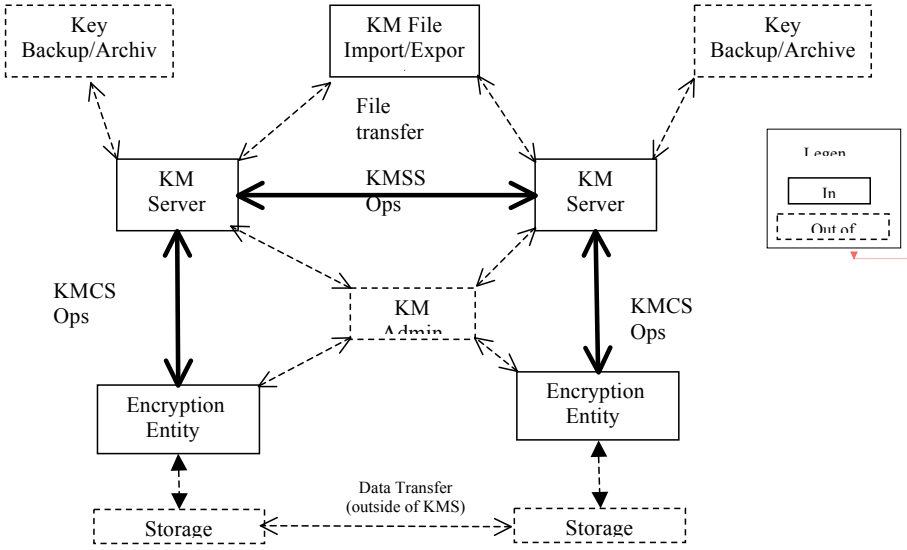


Figure 1— Key Management Architecture Model

The following list describes each component within Figure 1 in more detail:

— KM Server: The Key Management server is a software and/or hardware entity that is capable of generating, storing, protecting, and distributing cryptographic keys and other security related information necessary to support the operation of Encryption Entities. In addition, it may also support security services such has audit services, backup/archive services, security policy management, and access control services.

— Encryption Entity: An entity or collection of entities that is able to receive encryption keys, enforce policy, and encrypt or decrypt data to and from a storage medium (see 4.3).

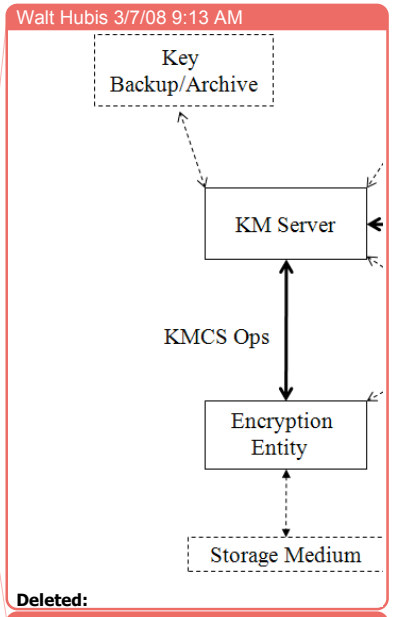
— Storage Medium: Any device that is used to store and retrieve data over extended periods of time in a persistent manner. This may include magnetic, optical, or solid-state memory devices (see 4.3).

— Key Backup and Archive: This function is performed by a software and/or hardware entity that provides secure, long-term storage of keys. This service may include key backup services for securing copies of currently active keying material, as well as key archive services to provide a repository for non-active key material of historical interest.

— KM File Import/Export: The Key Management File Import and Export service is a hardware and/or software entity that provides file exchange services between KM servers. This service will be defined in a subsequent revision of this specification.

— KM Admin: The key management administrator is an entity that is responsible for the configuration and management of the key management environment. There may be several KM Admins, but all must be authenticated with the management servers. The role, functions and authentication methods of the KM Admin is outside the scope of this document.

— KMSS Ops: The Key Management Server-Server Operations define the set of high level operations, their messages, and transport mechanisms that allow Key Management Servers to communicate with one another in order to facilitate the delivery of key management services that can not be accomplished with a single Key Management Server. Server-to-server operations facilitate advanced key



Walt Hubis 3/7/08 9:13 AM
Deleted:

Walt Hubis 3/7/08 9:17 AM
Deleted: ope

Walt Hubis 3/7/08 8:18 AM
Deleted: sloid

Walt Hubis 3/7/08 8:19 AM
Deleted: e

Walt Hubis 3/7/08 8:21 AM
Deleted: [WAH- Note sure about the purpose of this and wether we should state that key material is not transferred by this interface. Otherwise, we'll need to define the security for this service and associated interface(s)][WAH- Is this really in-scope?]

Walt Hubis 3/7/08 8:23 AM
Deleted: User

Walt Hubis 3/7/08 8:23 AM
Deleted: users

Walt Hubis 3/7/08 8:23 AM
Deleted: U

Walt Hubis 3/7/08 8:23 AM
Deleted: sers

Walt Hubis 3/7/08 8:24 AM
Deleted: e mkey

Walt Hubis 3/7/08 8:24 AM
Deleted: [WAH- Do we want to specify the authentication required by the KM Server for the KM User?]

management services such as hierarchical key distribution and redundancy and availability of key services for key management environments.

— KMCS Ops: The Key Management Client-Server Operations define the set of high level operations, their messages, and transport mechanisms that allow Key Management Clients to request key management services from Key Management Servers and, likewise, to allow Key Management Servers to respond to these key management service requests. These operations utilize a well defined set of key management objects for the communications of key management related information between the clients and servers.

— Scope: The KM Servers, KM Clients, Encryption Entities, KMSS Operations and KMCS Operations are the subject of this document; all other entities are included for completeness, but are nor defined in this documents.

Walt Hubis 3/7/08 8:29 AM
Deleted: Managemment

Matthew Ball 3/10/08 2:04 PM
Deleted:

Walt Hubis 3/7/08 8:29 AM
Deleted: [WAH- How about File Import/Export?]

4.3 Key Management Conceptual Model

The key management conceptual model is defined in this standard to provide a more detailed view of the relationship between the critical entities involved in the key management environment. The purpose of the key management conceptual model is fourfold:

- To explicitly identify the differences between data plane operations and control plane operations. The process of actually using an encryption key to encrypt and decrypt data occurs in the data plane. The process of key acquisition and activation occurs in the control plane.
- To identify the entities, components, and their interfaces which are defined as part of this standard.
- To identify the entities and components which are not within the scope of the standard, but need to be understood based on their impact to a key management environment.
- To understand possible embodiments of this standard in actual implementations. The key management conceptual model, coupled with the use cases defined in the informative annex of this standard will aid implementers in their efforts to be compliant to this standard.

Figure 2 illustrates the conceptual model for the key management environment defined in this standard.

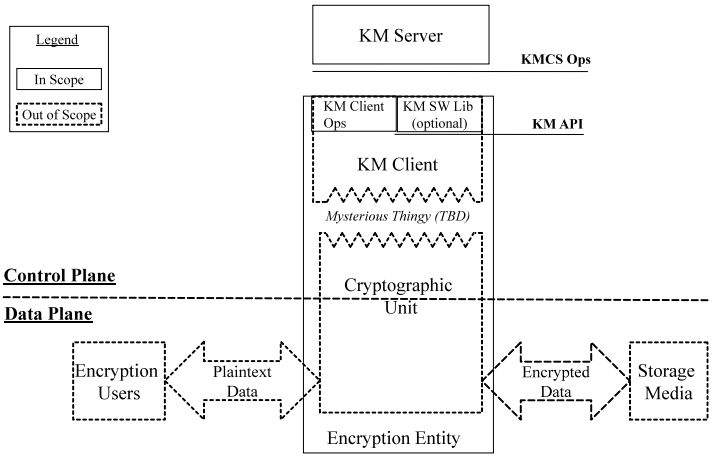


Figure 2— Key Management Conceptual Model

The following list describes each component within Figure 2:

— KM Server: See 4.2

— Control Plane: Where data set information, cryptographic keys, and security parameters are transported between the Cryptographic Unit and the Key Management Client

— Data Plane: Where data encryption and decryption actually take place

— Encryption Users: Defines all users, servers, applications, and systems that require the stored data media used for their persistent storage requirements be protected by encryption.

— Storage Media: Anything capable of storing non-volatile data.

— Encryption Entity: A entity that contains the following logical or physical components:

— KM Client (In Scope): A component of an Encryption Entity that serves several functions:

i) Communicates with Key Management Servers to acquire cryptographic keys;

ii) Communicates with Cryptographic Units to load and activate these keys for their use in encrypting and decrypting data that is written and read, respectively, to data storage media, and;

iii) Enforces the Key Management Policies as required in this specification.

The KM Client enforces these functions in one of two ways:

i) KM Client Ops: The key management client enforces these functions through the appropriate implementation and use of the KM Client Operations (out of scope in this document).

ii) KM SW Lib: These functions may be enforced through the use of a standard Key Management Software Library as defined in this document.

Walt Hubis 3/7/08 8:30 AM
Formatted: Bullets and Numbering

Walt Hubis 3/7/08 8:33 AM
Deleted: and,

Walt Hubis 3/7/08 8:33 AM
Deleted: ,

Walt Hubis 3/7/08 8:32 AM
Deleted: specifaction

Walt Hubis 3/7/08 8:35 AM
Formatted: IEEEStds Numbered List Level 3, Space After: 0 pt, Outline numbered + Level: 3 + Numbering Style: i, ii, iii, ... + Start at: 1 + Alignment: Left + Aligned at: 0.88" + Tab after: 1.25" +

Walt Hubis 3/7/08 8:37 AM
Deleted: can

Walt Hubis 3/7/08 8:32 AM
Deleted: doecument

Walt Hubis 3/7/08 8:37 AM
Deleted: , or

— Cryptographic Unit (out of scope): An entity capable of using cryptographic algorithms to encrypt data being written to and decrypt data read from storage media.

Walt Hubis 3/7/08 8:38 AM

Deleted: Cryptographic units implement the cryptographic algorithms specified in IEEE P1619, IEEE P1619.1, and IEEE P1619.2. A Cryptographic Unit lies in both the data plane (where encryption and decryption actually take place) and the control plane (where data set information, cryptographic keys, and security parameters are transported between itself and the Key Management Client.

4.4 Key Lifecycle Model

A crucial aspect of this key management standard is the concise definition for the states of encryption keys and the various transitions that can occur between these states. This definition of states of transitions between states for encryption keys is commonly referred to as the key lifecycle. Several standards bodies have attempted to define general key lifecycle models with their respective key state and transition definitions. These models were typically created in the context of using keys for protecting “data in flight” over communications links. The ephemeral nature of the keys and protected communication “sessions” does not lend itself well to the using these models “as-is” in the protection of stored data, or “data at rest” as it is commonly known.

A modified key lifecycle model is required that adequately define the key states and transitions between these states to ensure the highest levels of protection for stored data. The key lifecycle model defined herein is based largely on the key lifecycle models defined in NIST SP800-57 Part 1, and IEC/ISO 11770-1, but with additional states, transitions, and clarifications of these states and transitions in context of key management for the protection of stored data environments. The informative annex of this standard provides a concise comparison of the key lifecycle states and transitions presented in this standard, and compares them to the key lifecycle states and transitions presented in the key lifecycle models of applicable international and national standards.

Figure 3 illustrates the key lifecycle model as defined in this standard.

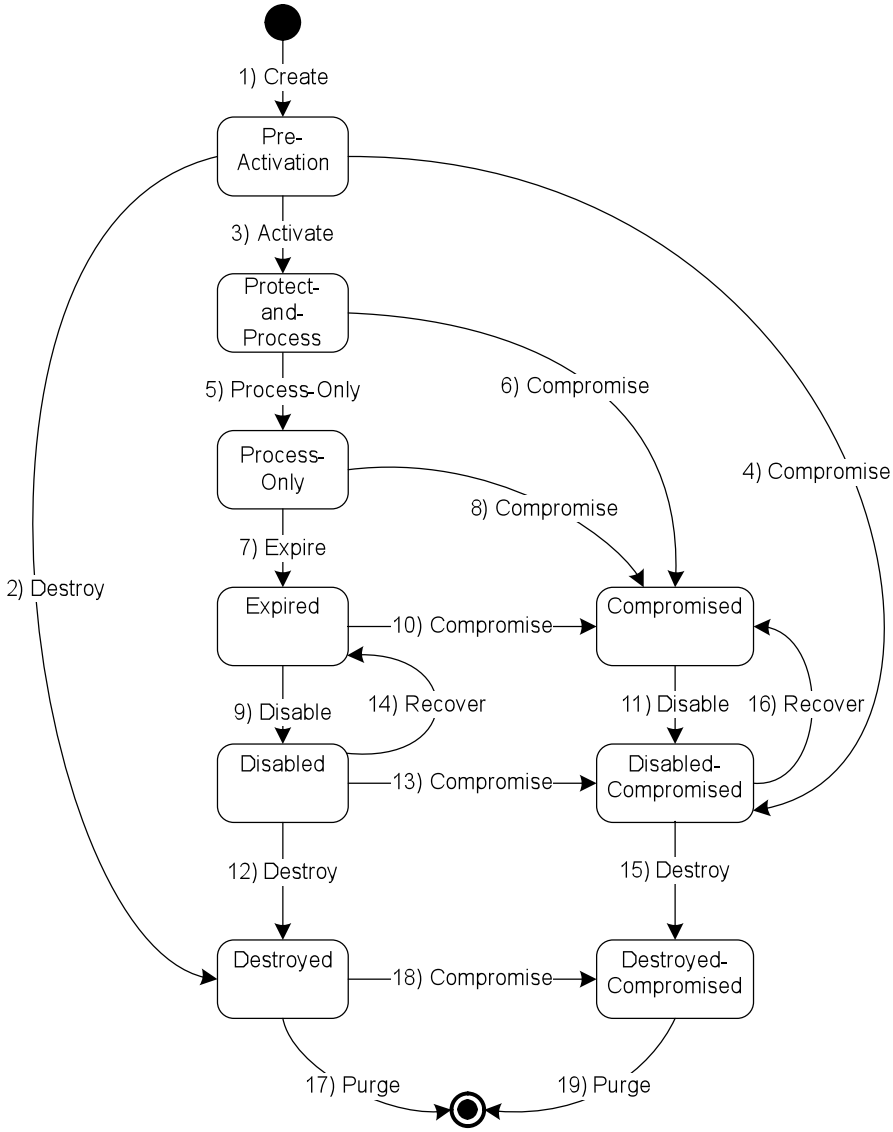


Figure 3— Key Lifecycle Model

The key lifecycle model consists of a set of key states in which depicts the states any particular encryption key can exist in at any particular time. In addition, this model indicates the allowable transitions between states. The definitions of these transitions include the events or actions necessary to cause a key to transition from one state to the next. These events or actions include time out events or user/management controlled actions. To complete the key lifecycle model, the definition of a set of time periods must be

defined that control the automated transition of keys between several of the states in the key lifecycle model.

4.4.1.1 Key State Definitions

The following are the formal descriptions of the key states defined in the key lifecycle model:

— **Pre-Activation**: The key has generated but is not yet in use or distributed to any KM Client or Cryptographic Unit. A key in this state can only exist in the KM Server. Such a key can be distributed to a KM Client.

— **Protect-and-Process**: A key in this state can be used for both encryption (i.e., to protect information) and decryption (to process information). A key is placed into this state when it is initially given to a KM Client, i.e., it is activated. The activation is done when a KM Client requests a new key. If a key is created by a KM Client or Cryptographic Unit and is then given to the KM Server, the key will be immediately placed into Protect-and-Process state. A key will remain in this state until the encryption period passes.

— **Process-Only**: A key in this state can be used for decryption but not encryption. When a KM Client/Cryptographic Unit determines that none of the keys available to it (e.g., for a specific tape cartridge or disk volume that is being read or written) are in the protect-and-process state, and it needs to perform encryption, it should create a new key. Keys transition from Protect-and-Process to Process-Only when the Encryption Period for the key expires, or as a result of an administrative action. A key will remain in the Process-Only state until the Crypto Period passes. At that time the key will be expired and move to the Expired state.

— **Expired**: An expired key is a key that has had its Crypto Period expire, but it may still be needed to process (decrypt) information. Keys in this state may be used to process (decrypt) data only. An expired key can be delivered to a KM Client, but the Cryptographic Unit should not use the key to encrypt data. A key will remain in this state until the key's Disable Period expires or as a result of an administrative action. At that time the key will be disabled and it will move to the Disabled state.

  The difference between keys in the Process-Only and Expired states is subtle. As far a KM Clients or Cryptographic Units are concerned, the states are equivalent. The difference is significant only to KM Servers. A key in the Process-Only key is still in the NIST operational phase and would routinely be delivered to KM Clients. A key which has had its Crypto Period expire has moved into the NIST post-operational phase, and may have additional restrictions imposed on its delivery to KM Clients.

— **Disabled**: A disabled key is a key that is still known to the KM Server but it will not be delivered to a KM Client. A disabled key's key material remains intact in the KM Server. A key will remain in the Disabled state until either the key's Destruction Period expires, or administrative action places the key in the Destroyed state, Compromised state, or recovers the key to the Expired state. If the key's Destruction Period expires, the key will be destroyed and will transition to the Destroyed state.

— **Compromised**: Keys are compromised when they are released to or discovered by an unauthorized entity. Compromised keys should not be used to protect information, but may need to be used to process information. The KM Server cannot determine if a key has been compromised. An administrative action is needed to inform a KM server that a key has been compromised. A compromised key will be delivered to a KM Client, but the Cryptographic Unit should not use the key to encrypt data. Just as with Expired keys, a KM Server may impose additional restrictions on the delivery of compromised keys to a KM Client. A key will remain in this state until the Disable Period expires. At that time the key will be disabled and move to the Disabled-Compromised state.

— **Disabled-Compromised**: A key that is both disabled and compromised. Disabled and compromised keys will never be delivered to KM Clients. A key will remain in the disabled compromised state until either the Destruction Period expires or the key is recovered to the Compromised state by

administrative action. If the Destroy Period expires, the key will be destroyed and moved to the Destroyed-Compromised state.

— **Destroyed**: A destroyed key is a key which has had its key material removed from the KM Server. Information or metadata about the key may be retained by the KM Server. Destroyed keys cannot be delivered to a KM Client. Keys can be destroyed by either an administrative action in the KM Server, or by the expiration of the Destruction Period of the key while it is in the Disabled or Disabled-Compromised states.

— **Destroyed-Compromised**: Similar to keys in the Destroyed state, but the key was compromised before or after destruction.

— **Terminal (Purged)**: Purged is the terminal state for keys. A purged key is a key that no longer exists in the KM Server in any form. Neither the key material nor any metadata about the key is known to the KM Server. A purged key cannot be delivered to a KM Client.

4.4.1.2 Key State Transition Definitions

The following are the formal descriptions of the key state transitions defined in the key lifecycle model:

— **(1) Create**: When a key is created by a KM Server, it begins in the pre-activation state. This transition occurs as soon as a key is generated within the KM Server, such as a key being generated by a random number generator (RNG).

Transition 1 applies to newly created keys; the key immediately transitions to Pre-Activation state upon its creation.

— **(2) Destroy**: A key in the Pre-Activation state may be moved directly to the Destroyed state. If the key has never been activated and is no longer required, but there is a requirement to maintain information about the key, the key may be destroyed. This transition can only occur as a result of administrative action.

Transition 2 applies to keys in the Pre-Activation state; the key transitions to the Destroyed state.

— **(3) Activate**: A key transitions from the Pre-Activation state to the Protect-and-Process state when it is available for use. This transition occurs when the key is assigned for use by a KM Client. There is a bit of confusion around the terminology, since a KM Client may view this action as “creating a key” even though the KM Server views this as assigning a previously generated key.

Transition 3 applies to keys in the Pre-Activation state; the key transitions to the Protect-and-Process state.

— **(4) Compromise**: A key transitions to the Compromised state when it has been determined to have been released to or discovered by an unauthorized entity. The KM Server cannot determine that this has happened, so this transition occurs as the result of an administrative action.

Transition 4 applies to keys in the Pre-Activation state; the key transitions to the Compromised state.

— **(5) Process-Only**: A key transitions from the Protect-and-Process state to the Process-Only state when a period of time equal the Encryption Period has expired after the key is activated. This transition may also occur as a result of administrative action. Some KM Clients and Cryptographic Units may be unable to strictly enforce this transition.

Transition 5 applies to keys in the Protect-and-Process state; the key transitions to the Process-Only state.

— **(6) Compromise**: Same actions as for the (4) Compromise transition.

Transition 6 applies to keys in the Protect-and-Process state; the key transitions to the Compromised state.

— **(7) Expire**: A key transitions from active to Expired when its Crypto Period expires. This transition may also occur as a result of administrative action.

  Transition 7 applies to keys in the Process-Only state; the key transitions to the Expired state.

— **(8) Compromise**: Same actions as for the (4) Compromise transition. Transition 8 applies to keys in the Process-Only state; the key transitions to the Compromised state.

— **(9) Disable**: This transition will occur after the disable period for a key passes. This transition can also occur as a result of administrative actions. The key material and its metadata will be retained by the KM Server. The key will no longer delivered to KM Clients.

  Transition 9 applies to keys in the Expired state; the key will transition to the Disabled state.

— **(10) Compromise**: Same actions as for the (4) Compromise transition.

  Transition 10 applies to keys in the Expired state; the key transitions to the Compromised state.

— **(11) Disable**: Same actions as for the (9) Disable transition.

  Transition 11 applies to keys in the Compromised state; the key transitions to the Disabled-Compromised state.

— **(12) Destroy**: A key can be destroyed when it is no longer needed. When the Destruction Period for a key expires, it will be destroyed. This transition can also occur as a result of administrative action. Active keys (keys in the Protect-and-Process or Protect-Only states) cannot be destroyed; only keys that have been previously activated and are now in the Disabled or Disabled-Compromised states can be destroyed. When a key is destroyed, the key material is removed from the KM Server. Metadata about the key is retained in the KM Server.

  Transition 12 applies to keys in the Disabled state; the key transitions to Destroyed state.

— **(13) Compromise**: Same actions as for the (4) Compromise transition.

  Transition 13 applies to keys in the Disabled state; the key transitions to Disabled-Compromised state.

— **(14) Recover**: When a disabled key is required to be delivered to KM Clients, it can be recovered. This occurs as a result of an administrative action.

  Transition 14 applies to key in the Disabled state; the key transitions to Expired state.

— **(15) Destroy**: Same actions as for the (12) Destroy transition.

  Transition 15 applies to key in the Disabled-Compromised state; the key transitions to the Destroyed-Compromised state.

— **(16) Recover**: Same actions as for the (14) Recover transition.

  Transition 20 applies to keys in the Disabled-Compromised state; the key transitions to the Compromised state.

— **(17) Purge**: The action of purging a key removes all information about the key from the KM Server's key records. Neither the key material nor the key value will be retained by the KM Server. Note, however, that audit logs or other indirect information in the KM Server may still contain information about the key.

  Transition 17 applies to keys in the Destroyed state; the key transitions to the Purged state. However, since the key does not exist in the KM Server, there is no record for the key showing the Purged state.

— **(18) Compromise**: Same actions as for the (4) Compromise transition.

  Transition 18) applies to keys in the Destroyed state; the key transitions to the Destroyed-Compromised state.

— **(19) Purge**: Same actions as for the (17) Purge transition.

Transition 19 applies to keys in the Destroyed-Compromised state; the key transitions to the Purged state. However, since the key does not exist in the KM Server, there is no record for the key showing the purged state.

4.4.1.3 Key Time Period Definitions

Many of the key states and transitions defined in the key lifecycle model are dependent on one or more time related attributes associated with the keys. These time related attributes are typically categorized as timer values or timeout periods that are activated when keys enter certain states and their expiration cause transitions to certain states.

The key life cycle is based on four time periods:

- **Encryption Period**: This is the period of time after a key is activated that it can be used to encrypt data.
- **Crypto Period**: This is the period of time after a key is activated that it can be used for routine decryption. This time period should always be as long as or longer than the Encryption Period.
- **Disable Period**: This is the period of time after a key is activated before a key will be disabled and become unavailable to KM Clients. A key may still be delivered to a KM Client after the Crypto Period ends but before the Disable Period ends, however, this may require administrative action on the KM Server. This case would be for a non-routine usage of the key. This time period should always be as long as or longer than the Crypto Period.
- **Destruction Period**: This is the period of time after a key is activated before the key is destroyed by the KM Server. This time period should always be as long as or longer than the Disable Period.

All key related time periods start when a key is activated; this occurs when a key is given to a KM Client by a KM Server, or when a key generated by a KM Client is given to the KM Server. Any time period can range from zero to forever, with the limitation that each time period must be at least as long as its predecessor as specified above.

Figure 4 illustrates the key related time periods and their relationship to the keys states of the key lifecycle model.

{mention that these are not the only time periods available – these are the time periods that can cause a key state transition.}

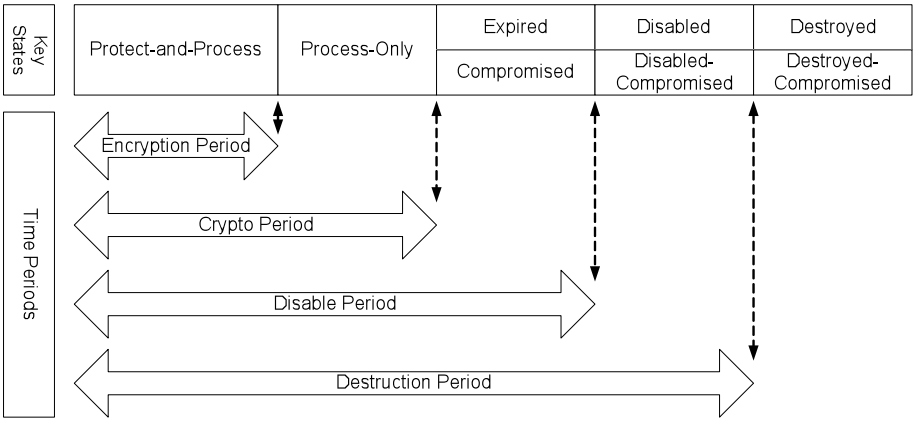


Figure 4— Key Time Periods Compared to Key States

Matthew Ball 3/10/08 1:51 PM
Formatted: Bullets and Numbering

4.5 Key Management Sequence Models

{Content TBD}

Walt Hubis 3/7/08 8:14 AM
Formatted: Bullets and Numbering

Matthew Ball 3/10/08 1:52 PM
Formatted: IEEEStds Paragraph

4.6 Key Management Operations Model

Another important aspect of this key management standard is the conformant operations that are defined to take place between a Key Management Client (KM Client) and a Key Management Server (KM Server). A key management operation model is presented here to illustrate the high level flow of management information between these KM Client and KM Server entities. The general sequence of operations, notions of synchronous versus asynchronous operations, and support for single threaded versus multi-threaded operations are critical the interoperability of conformant implementations of this standard.

More text and diagrams in development.

Walt Hubis 3/7/08 8:14 AM
Formatted: Bullets and Numbering

4.7 Key Management Object Model

To properly perform key management operations, a series of key management objects and their relationships must be defined to ensure interoperability of conformant implementations of this standard. Key management objects include all data structures and information that are necessary to provide complete key management services for encryption of stored data applications.

[Need D1 Word source to copy object model definitions here.]

Matthew Ball 3/10/08 1:50 PM
Formatted: IEEEStds Level 1 Header, None, Space Before: 0 pt, After: 0 pt, Don't keep lines together, Hyphenate

5. Namespace

{Move Namespace subclause here}

Matthew Ball 3/10/08 1:49 PM
Formatted: Bullets and Numbering

Matthew Ball 3/10/08 1:50 PM
Deleted:

Matthew Ball 3/10/08 1:51 PM
Formatted: IEEEStds Paragraph, None, Space Before: 0 pt, After: 0 pt, Don't keep lines together, Hyphenate

6. Key Management Operations

[Any changes required will be coordinated with Operations and Object subcommittee's proposals.]

7. Key Management Transport

[Any changes required will be coordinated with Transport subcommittee proposals.]

Walt Hubis 3/7/08 8:14 AM
Formatted: Bullets and Numbering

8. Key Management Messaging

Walt Hubis 3/7/08 8:14 AM
Formatted: Bullets and Numbering

[Any changes required will be coordinated with Messaging subcommittee proposals.]

-
- 8 Requirements for KM Servers (shalls)
- 9 Requirements for KM Clients (shalls)
- 10 Requirements for KM SW Lib (shalls)

Annex A

(informative)

Bibliography

[List all bibliographic material here]

Annex B

(informative)

Example Use Cases

[Objects & operations or use cases should add the appropriate use cases here]

Annex C

(informative)

XML and TLV Schema Definitions

C.1 XML Schema

[Additional messaging group information for selected XML syntax goes here]

C.2 TLV Schema – {No such thing}

[Additional messaging group information for selected TLV schema goes here]
ASN.1 DER
Freeform

Annex D

(informative)

Comparison of P1619.3 Key Lifecycle Model with Other Standards

D.1 Key State Comparisons

Table D.1 compares the mappings from the P1619.3 key states to the key states defined in the NIST SP 800-57 lifecycle model and the IEC/ISO 11770-1 lifecycle model:

P1619.3 Key State	NIST SP 800-57 Key State	IEC/ISO 11770-1 Key State	Notes
Pre-Activation	Pre-activation	Pending Active	Notes 1 & 2
Protect-and-Process	Active	Active	Notes 3 & 4
Process-Only	Active	Active	Notes 3 & 4
Expired	Deactivated	Deactivated	Notes 3 & 4
Disabled	Deactivated	Deactivated	Notes 3 & 4
Compromised	Compromised	None	Notes 3 & 5
Disabled-Compromised	Compromised	None	Notes 3 & 5
Destroyed	Destroyed	Destroyed (Implied)	Notes 1 & 6
Destroyed-Compromised	Destroyed Compromised	None	Notes 1 & 5
Purged	None	None	Notes 7 & 5

Table D.1 — Standard Key State Comparisons

Note 1: IEEE P1619.3 key state is identical to the key state defined in the NIST SP 800-57 standard.
Note 2: IEEE P1619.3 key state is identical to the key state defined in the IEC/ISO 11770-1 standard.
Note 3: IEEE P1619.3 key state is a substate of the key state defined in the NIST SP 800-57 standard.
Note 4: IEEE P1619.3 key state is a substate of the key state defined in the IEC/ISO 11770-1 standard.
Note 5: IEEE P1619.3 key state is a new state not defined in the IEC/ISO 11770-1 standard.
Note 6: IEEE P1619.3 key state is the same as the state implied in the IEC/ISO 11770-1 standard.
Note 7: IEEE P1619.3 key state is a new state not defined in the NIST SP 800-57 standard.

D.2 Key Transition Comparisons

[Need Jon Holman and/or Larry Hofer to help complete this subclause.]

Table D.2 compares the mappings from the P1619.3 key transitions to the key transitions defined in the NIST SP 800-57 lifecycle model and the IEC/ISO 11770-1 lifecycle model:

P1619.3 Key Transition	NIST SP 800-57 Key Transition	IEC/ISO 11770-1 Key Transition	Notes
Create	Transition 1	Generation	Notes 1 & 2
Destroy	Transition 2	Destruction	Notes 1 & 2
Activate	Transition 4	Activation	Notes 1 & 2
Compromise	Transition 3	None	Notes 1 & 4
Process-Only	None	None	Notes 3 & 4
Compromise	Transition 5	None	Notes 1 & 4
Expire	Transition 6	Deactivation	Notes 1 & 2
Compromise	Transition 5	None	Notes 1 & 4
Disable	None	None	Notes 3 & 4
Compromise	Transition 8	None	Notes 1 & 4
11) Disable	None	None	Notes 3 & 4
12) Destroy	Transition 7	Destruction	Notes 1 & 2
13) Compromise	Transition 8	None	Note 1 & 4
14) Recover	None	Reactivation (see note)	Notes 5 & 6
15) Destroy	Transition 9	Destruction	Notes 1 & 2
16) Recover	None	Reactivation (see note)	Notes 5 & 6
17) Purge	None	None	Notes 3 & 4
18) Compromise	Transition 10	None	Note 1 & 4
19) Purge	None	None	Note 3 & 4

Table D.2 — Standard Key Transition Comparisons

Note 1: IEEE P1619.3 key state transition is identical to the key state transition defined in the NIST SP 800-57 standard.
Note 2: IEEE P1619.3 key state transition is identical to the key state transition defined in the IEC/ISO 11770-1 standard.
Note 3: IEEE P1619.3 key state transition is a new key state transition not defined in the NIST SP 800-57 standard.
Note 4: IEEE P1619.3 key state transition is a new key state transition not defined in the IEC/ISO 11770-1 standard.
Note 5: IEEE P1619.3 key state transition is implied by the discussion in the NIST SP 800-57 standard, but is not explicitly shown.
Note 6: IEEE P1619.3 key state transition is similar to the Reactivation transtion described in IEC/ISO 11770-1 standard, but does not result in the key transition back to the active state.

Walt Hubis 3/7/08 8:58 AM
Formatted Table ... [1]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [2]
Walt Hubis 3/7/08 8:52 AM
Deleted: TBD
Walt Hubis 3/7/08 8:57 AM
Formatted ... [3]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [4]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [5]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [6]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [7]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [8]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [9]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [10]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [11]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [12]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [13]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [14]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [15]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [16]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [17]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [18]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [19]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [20]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [21]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [22]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [23]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [24]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [25]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [26]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [27]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [28]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [29]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [30]
Walt Hubis 3/7/08 8:57 AM
Formatted ... [31]
Walt Hubis 3/7/08 9:03 AM

D.3 Key Time Period Comparisons

The P1619.3 key life cycle time periods are based on the time periods defined in NIST SP 800-57; however, this standard renames the time periods, places limits on their relationships, and adds two additional time periods. These changes were necessary to support the concept of long term key usage required for encryption of stored data applications.

Walt Hubis 3/7/08 9:05 AM
Deleted: *[Need Jon Holman and/or Larry Hofer to help complete/check this subclause.]*

Figure D.1 illustrates the key related time periods defined in the P1619.3 standard. Figure D.2 provides a visual depiction of the key related time periods defined in NIST SP 800-57.

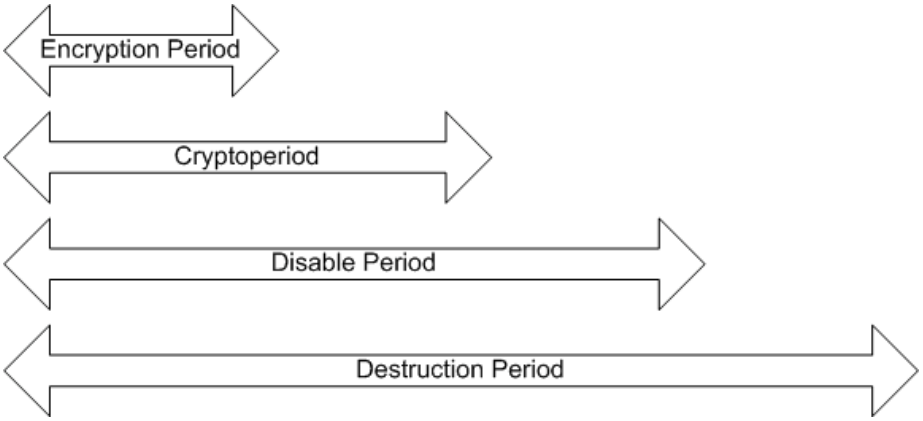


Figure D.1— P1619.3 Key Related Time Periods

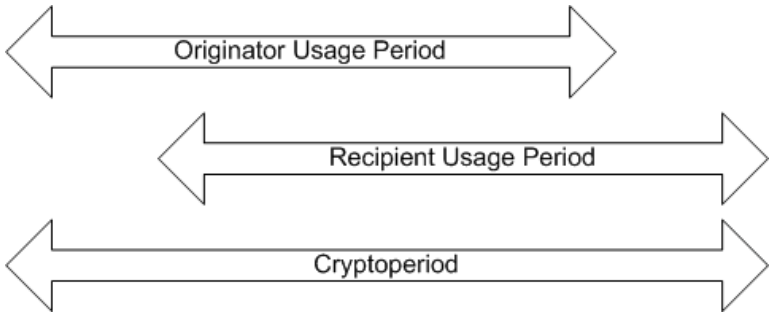


Figure D.2— NIST SP 800-57 Key Related Time Periods

The first two time periods, illustrated above, in the P1619.3 standard are a simplification from NIST's terminology and approach of providing two time periods that may be offset. NIST defines the "originator usage period" and "recipient usage period".

The NIST's originator usage period is the same as P1619.3 standard's Encryption Period. NIST's recipient usage period might start after the originator usage period. Since storage devices might need to read data implicitly as soon as it's written, the simplified approach utilized by P1619.3 is used. Therefore, the simplified P1619.3 diagram is more appropriate for stored data encryption keys.

Walt Hubis 3/7/08 9:06 AM
Deleted: ay

Walt Hubis 3/7/08 9:06 AM
Deleted: ay

The last two P1619.3 key related time periods (Disable Period and Destruction Period) are not explicitly defined in NIST SP 800-57 because they are directly applicable to encryption of stored data and are not required the ephemeral nature of data protected by encryption of communication links.