SP 800-130

A Framework for Designing Cryptographic Key Management Systems

5/25/2012 Lunch and Learn

Scott Shorter

Topics

- Follows the Sections of SP 800-130 draft 2:
  - Introduction
  - Framework Basics
  - Goals of a CKMS
  - Security Policies
  - Roles and Responsibilities
  - Keys and Metadata
  - Interoperability and Transitioning
  - Security Controls
  - Testing and System Assurance
  - Disaster Recovery
  - Security Assessment
  - Technological Challenges

SP 800-130 Metadata

- **Title:** A Framework for Designing Cryptographic Key Management Systems
- **Revision:** draft 2
- **Date:** 3/21/2012
- **Authors:** Elaine Barker, Miles Smid, Dennis Branstad, Santosh Chokhani

Author Accomplishments

- **Barker:**
  - SP 800-57 Recommendations on Key Management
  - SP 800-90A Recommendation for Random Number Generation Using Deterministic Random Bit Generators
- **Branstad:**
  - NIST SP 800-79-1, Guidelines for the Accreditation of Personal Identity Verification Card Issuers
- **Chokhani:**
  - IETF RFCs 2527 & 3647 on Internet X.509 Public Key Infrastructure Certificate Policy and Certification Practices Framework
- **Smid:**
  - Retired from NIST CSD
  - Godfather of FIPS 140 and AES
  - Co-author of NIST SP 800-57 series.

Section 1 – Introduction

- Cryptography is used:
  - To protect information from unauthorized disclosure
  - to detect unauthorized modification, and
  - to authenticate the identities of system entities
- Cryptography is useful:
  - when data transmission or authentication happens over channels that are not physically protected
  - As an additional layer of protection against insiders and hackers.

Cryptographic Services

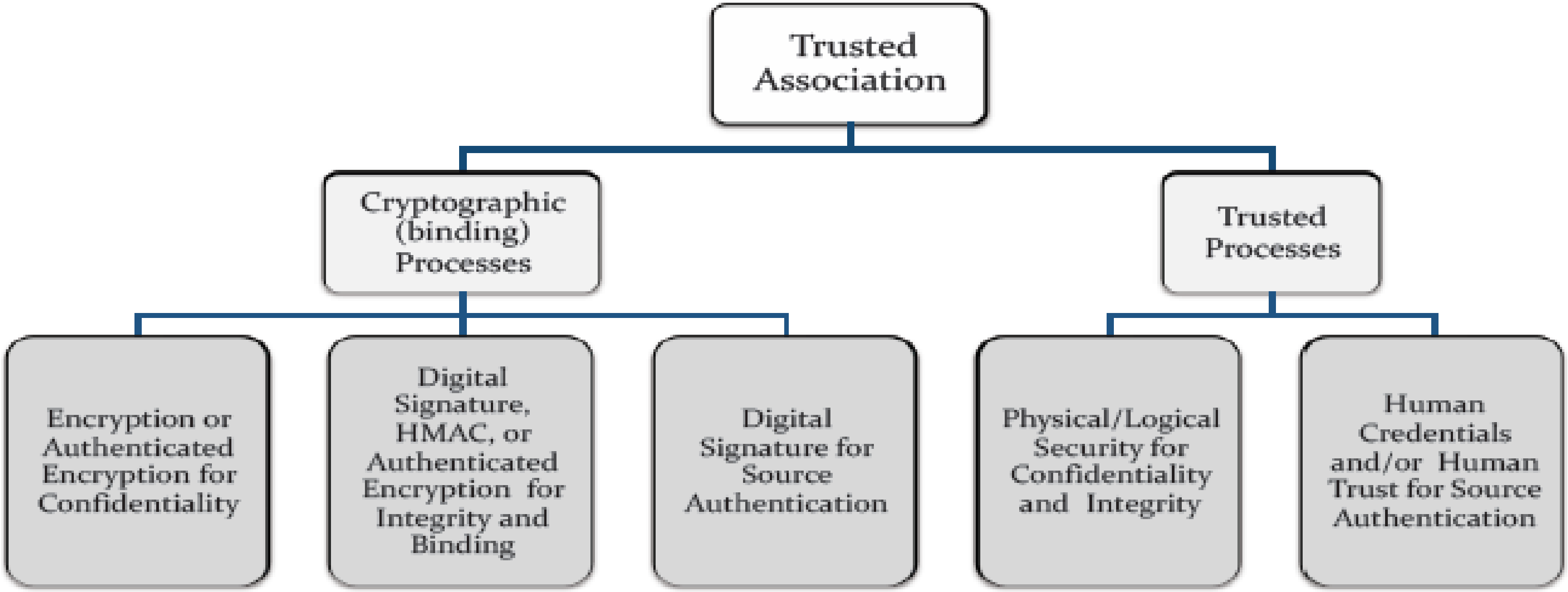
- Cryptography can provide the following types of protection
  - *Confidentiality* is obtained by encrypting data in such a way that only authorized parties can access the keys to decrypt the data.
  - *Integrity* can be demonstrated through digital signature or message authentication algorithms
  - *Source authentication* provides assurance that protected data came from an authorized entity.
- These protections are applicable to data, but also to cryptographic keys and metadata.
  - Example: X.509 certificates bind a public key to an individual or device identity. The certificate provides integrity protection to the public key, identity and other metadata, and source authentication of the issuer is derived from the signature on the certificate. This is called a *trusted association* in the framework

CKMS Defined

- Cryptographic Key Management System (CKMS)
  - The policies, procedures, components and devices that protect, manage and distribute cryptographic keys and associated metadata.
- CKMS Component
  - Any hardware, software, or firmware that is used to implement a CKMS.
- CKMS Device
  - Any combination of CKMS components that serve a specific purpose (e.g., firewalls, routers, transmission devices, cryptographic modules, and data storage devices).

Trusted Associations

- Trusted Association
  - The linking of a key with selected metadata elements so as to provide assurance that the key and its metadata are properly associated, originate from a particular source, have not been modified, and have been protected from unauthorized disclosure.


Trusted Association
Cryptographic (binding) Processes
Trusted Processes
Encryption or Authenticated Encryption for Confidentiality
Digital Signature, HMAC, or Authenticated Encryption for Integrity and Binding
Digital Signature for Source Authentication
Physical/Logical Security for Confidentiality and Integrity
Human Credentials and/or Human Trust for Source Authentication

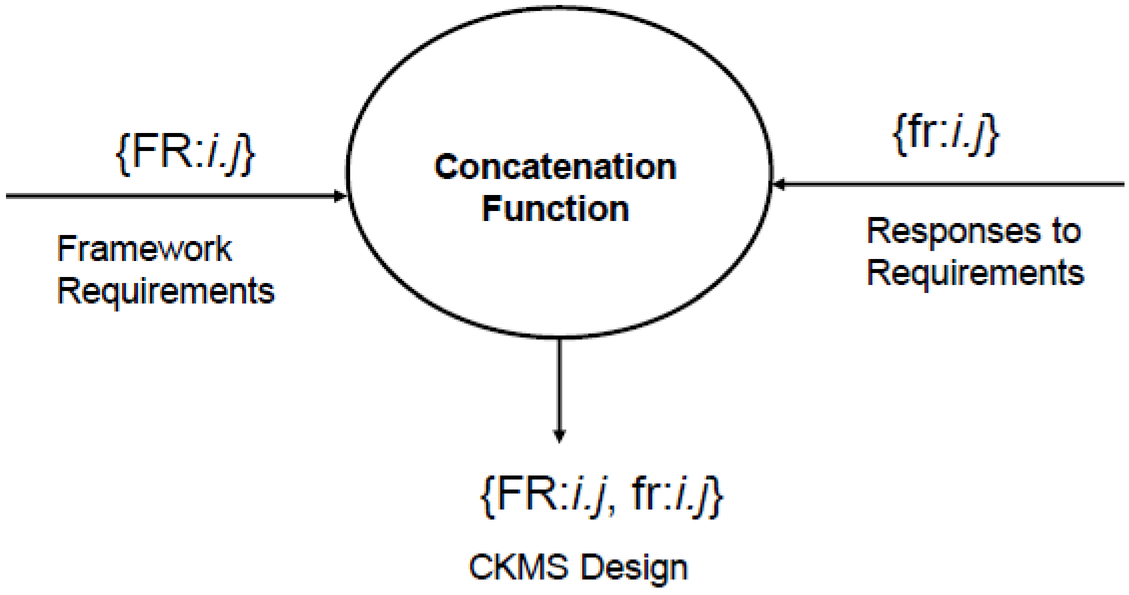
Framework Requirements

- The document contains a number of Framework Requirements (FRs) identified as "**FR:*i.j***", which indicates the *j*th FR in Section *i.*
- These are documentation requirements to be addressed when designing a CKMS

- **FR:1.1** A conformant CKMS design **shall** meet all "**shall**" requirements of the Framework.

CKMS Design Process

The CKMS Design consists of the Framework Requirements combined with the reponses to those requirements by the developers of the CKMS.



So the framework requirements are basically a set of questions that once answered by the CKMS developer will produce a complete CKMS design.

Section 2 – Framework Basics

- “This Framework does not impose any specific policies, procedures, security requirements, or system design constraints on the CKMS; it simply requires that they be documented in a structured manner so that various CKMS designs can be understood and compared.”

- So the framework provides a way to document CKMS designs in a common manner, not policy stipulations.

Algorithms and security strengths

- **FR:2.1** The CKMS design **shall** specify all cryptographic algorithms and supported key sizes for each algorithm used by the system.
- **FR:2.2** The CKMS design **shall** specify the estimated security strength (measured in bits of security) of all the cryptographic mechanisms that are employed to protect keys and their bound metadata.

Requirements and devices

- **FR:2.3** A compliant CKMS design **shall** make selections and provide documentation as required by the requirements the Framework.
- **FR:2.4** The CKMS design **shall** specify (e.g., make, model, and version) all major devices of the CKMS.

Section 3 - Goals of the CKMS

- **FR:3.1** The CKMS design **shall** specify its goals with respect to the communications networks on which it will function.
- **FR:3.2** The CKMS design **shall** specify the intended applications that it will support.
- **FR:3.3** The CKMS design **shall** list the intended number of users and the responsibilities that the CKMS places on those users.

Goals: Standards Conformance

- **FR:3.4 The** CKMS design **shall** specify the Federal, national, and international standards that are utilized by the CKMS and how conformance is tested for each.
- **FR:3.5** The CKMS design **shall** specify which commercial products are utilized in the CKMS design.
- **FR:3.6** The CKMS design **shall** specify all security standards to which the CKMS conforms.

Goals: Usability

- **FR:3.7** The CKMS design **shall** specify all user interfaces to the system.
- **FR:3.8** The CKMS design **shall** specify the results of any user-acceptance tests that have been performed regarding the ease of using the proposed user interfaces.
- **FR:3.9** The CKMS design **shall** specify the design principles of the user interface.
- **FR:3.10** The CKMS design **shall** specify all human error-prevention or failsafe features designed into the system.

Goals: Performance Requirements

- **FR:3.11** The CKMS design **shall** specify the performance characteristics of the CKMS, including the average and peak workloads that can be handled for the types of functions and transactions implemented, and the response times for the types of functions and transactions under those respective workloads.
- **FR:3.12** The CKMS design **shall** specify the techniques used to scale the system to increased-workload demands.
- **FR:3.13** The CKMS design **shall** specify the extent to which the CKMS can be scaled to meet increased-workload demands. This **shall** be expressed in terms of additional workload, response times for the workload, and cost.

Goals: Maximize COTS

- **FR:3.14** The CKMS design **shall** specify the COTS products used in the CKMS.
- **FR:3.15** The CKMS design **shall** specify which security functions are performed by COTS products.
- **FR:3.16** The CKMS design **shall** specify how COTS products are configured and augmented to meet the CKMS goals.

Section 4 - Security Policies

- **Information Management Policy**
  - An Information Management Policy specifies what information is to be collected or created, and how it is to be managed.
- **Information Security Policy**
  - An Information Security Policy is created to support and enforce portions of an Information Management Policy by specifying in more detail what information is to be protected from anticipated threats and how that protection is to be attained.
- **CKMS Security Policy**
  - A CKMS Security Policy is created to establish and specify requirements for protecting the confidentiality, integrity, availability, and source authentication of all cryptographic keys and metadata used by the organization.

Information Management Policy

- identifies management roles and responsibilities
- establishes the authorization required for people performing these information management duties
- specifies what information is to be considered valuable and sensitive and how it is to be protected
  - what categories of information need to be protected against unauthorized disclosure, modification or destruction.
- foundation for an information security policy and dictate the levels of confidentiality, integrity, availability, and source authentication protections that must be provided for various categories of sensitive and valuable information

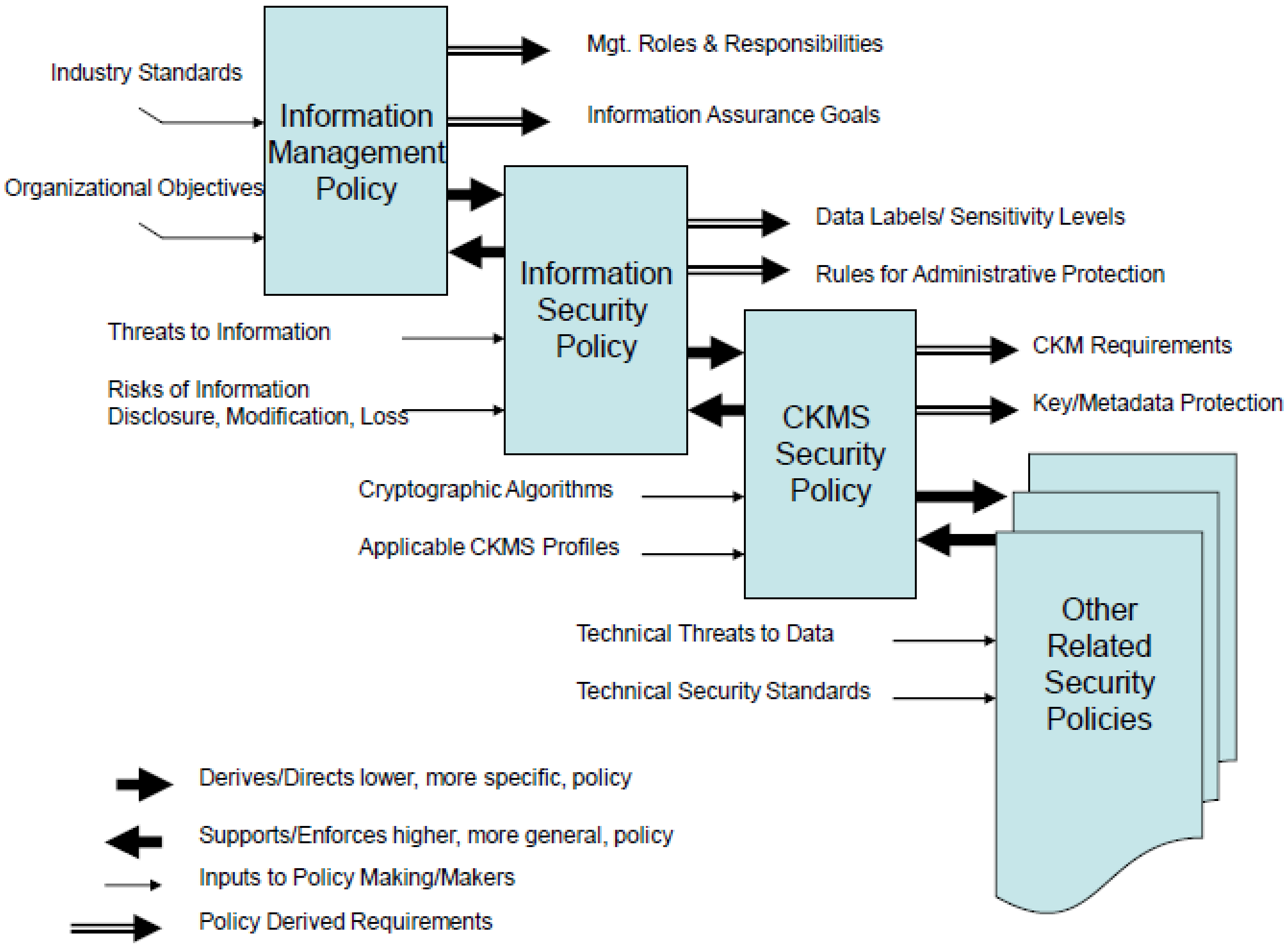
Information Security Policy

- The rules for collecting, protecting, and distributing valuable and sensitive information in both paper and electronic form are specified in this layer of policy.
- Inputs: Information Management Policy, potential threats and risks
- Outputs: information sensitivity levels and high-level rules for protecting information

CKMS Security Policy

- The selection of all cryptographic mechanisms and protocols to be used throughout the organization's automated information systems
- Includes the Key Security Policy, which identifies protections applied to each key type and its metadata over their entire lifecycle.
- Includes Key and Metadata Retention Policy enforced by the CKMS
- Includes justification for how it supports the Information Security Policy

Security Policies


Information Management Policy
Industry Standards
Organizational Objectives
Mgt. Roles & Responsibilities
Information Assurance Goals
Information Security Policy
Data Labels/ Sensitivity Levels
Rules for Administrative Protection
Threats to Information
Risks of Information Disclosure, Modification, Loss
CKMS Security Policy
CKM Requirements
Key/Metadata Protection
Cryptographic Algorithms
Applicable CKMS Profiles
Other Related Security Policies
Technical Threats to Data
Technical Security Standards
Derives/Directs lower, more specific, policy
Supports/Enforces higher, more general, policy
Inputs to Policy Making/Makers
Policy Derived Requirements

Specification of Security Policy

- **FR:4.1** The CKMS design **shall** specify the CKMS Security Policy that it enforces.
- **FR:4.2** The CKMS design **shall** specify how the CKMS Security Policy is to be enforced by the CKMS (e.g., the mechanisms used to provide the protection required by the policy).
- **FR:4.3** The CKMS design **shall** specify how any automated portions of the CKMS Security Policy are expressed in an unambiguous tabular form or a formal language (e.g., XML or ASN.1), such that an automated security system (e.g., table driven or syntax directed software mechanisms) in the CKMS can enforce them.
- **FR:4.4** The CKMS design **shall** specify other related security policies that support the CKMS Security Policy.
  - For example FIPS 140-2 security policy for cryptographic modules in the CKMS.

Key Sharing and Accountability

- **FR:4.5** The CKMS design **shall** specify the policies that describe the conditions under which keys and their metadata may be shared by two or more entities.
- **FR:4.6** The CKMS design **shall** specify how accountability is enforced by the CKMS.

Anonymity

- **FR:4.7** The CKMS design **shall** specify the anonymity, unlinkability and unobservability protection policies supported, and enforced by the CKMS.
- **FR:4.8** The CKMS design **shall** specify which CKMS transactions have or can be provided with anonymity protection.
- **FR:4.9** The CKMS design **shall** specify how CKMS transaction anonymity is achieved when anonymity is provided.
- **FR:4.10** The CKMS design **shall** specify which CKMS transactions have or can be provided with unlinkability protection.
- **FR:4.11** The CKMS design **shall** specify how CKMS transaction unlinkability is achieved.
- **FR:4.12** The CKMS design **shall** specify which CKMS transactions have or can be provided with unobservability protection.
- **FR:4.13** The CKMS design **shall** specify how CKMS transaction unobservability is achieved.

Other Security Domains

- **FR:4.16** The CKMS design **shall** specify the confidentiality, integrity, and source authentication policies that it enforces when communicating with entities from other security domains.
- **FR:4.17** The CKMS design **shall** specify what assurances it requires when communicating with entities from other security domains.
- **FR:4.18** The CKMS design **shall** specify its requirements for reviewing and verifying the security policies of other security domains with which it intends to communicate.
- **FR:4.19** The CKMS design **shall** specify how it prevents or at least warns an entity of the possible security consequences of communicating with an entity in a security domain with weaker policies**.**
- **FR:4.20** The CKMS design **shall** specify its policies regarding third-party sharing.

Multi-Level Security

- **FR:4.21** The CKMS design **shall** specify whether or not it supports multi-level security domains.
- **FR:4.22** The CKMS design **shall** specify each level of security domain that it supports.
- **FR:4.23** The CKMS design **shall** specify how it maintains the separation of the data belonging to each security level.
- **FR:4.24** The CKMS design **shall** specify whether or not it permits the upgrading or downgrading of data.
- **FR:4.25** The CKMS design **shall** specify how upgrading or downgrading capabilities are restricted to the domain security officer.

Section 5 – Roles and Responsibilities

- **System Authority:** A system authority is responsible to executive-level management (e.g., Chief Information Officer) for the overall operation and security of a CKMS. A system authority manages all operational CKMS roles. An operational role is a role that directly operates the CKMS.
- **System Administrator:** System administrators are responsible for the personnel, daily operation, training, maintenance, and related management of a CKMS other than its keys.
- **Cryptographic Officer**: A cryptographic officer is authorized to perform cryptographic initialization and management functions on the cryptographic module.
- **Domain Authority:** A domain authority is responsible for deciding the conditions necessary for communicating with another security domain and assuring that they are met.
- **Key Custodian:** A key custodian is designated to distribute and/or load keys or key splits into a CKMS

Roles Continued

- **Key Owner:** A key owner is an entity (e.g., person, group, organization, device, or module) authorized to use a cryptographic key or key pair and whose identifier is associated with a cryptographic key or key pair.
- **System User:** System users employ the CKMS when key management functions are required to support an application. System users may be, and often are, key owners.
- **Audit Administrator:** An audit administrator is responsible for auditing all aspects of a CKMS to verify its security and authorized operation.
- **Registration Agent:** A registration agent is responsible for registering new entities and binding their key(s) to their identifiers and perhaps other selected metadata. The registration agent may also enter entities into a database that contains an entity key, the key identifier, and other metadata for each entity.
- **Key Recovery Agent:** A key recovery agent is allowed to recover keys from backup or archive storage after identity verification and authorization of the requesting entity
- **CKMS Operator:** A CKMS operator is authorized to operate a CKMS in place of the system administrator as directed by the system administrator.

Section 6 – Keys and Metadata

This section contains:

- A list of types of keys
- A list of types of metadata
- A discussion of key lifecycle states and transitions
- A list of key/metadata management functions
- Security consideration for keys and metadata in storage and during key establishment
- Access restrictions to key management functions
- A discussion of compromise recovery

Key Types

Key Types:

1) Private Signature Key
2) Public Signature Key
3) Symmetric Authentication Key
4) Private Authentication Key
5) Public Authentication Key
6) Symmetric Data Encryption/Decryption Key
7) Symmetric Key Wrapping Key
8) Symmetric RNG Key
9) Private RNG Key
10) Public RNG Key
11) Symmetric Master Key
12) Private Key Transport Key
13) Public Key Transport Key
14) Symmetric Key Agreement Key
15) Private Static Key Agreement Key
16) Public Static Key Agreement Key
17) Private Ephemeral Key Agreement Key
18) Public Ephemeral Key Agreement Key
19) Symmetric Authorization Key
20) Private Authorization Key
21) Public Authorization Key

Possible Metadata 1

- **Key Label:** A key label is a text string that provides a human-readable and perhaps machine-readable set of descriptors for the key.
- **Key Identifier:** This element is used by the CKMS to select a specific key from a collection of keys. A key identifier is generally unique in a security domain.
- **Owner Identifier:** This element specifies the identifier (or identifiers) of the entity (or entities) that owns (or own) the key.
- **Key Life Cycle State:** A key life-cycle state is one of a set of finite states that describe the permitted conditions of a cryptographic key
- **Key Format Specifier:** This element is used to specify the format for the key.

Possible Metadata 2

- **Product used to create the Key:** This element specifies which cryptographic product was used to create or generate the key.
- **Cryptographic Algorithm using the Key:** This element specifies the cryptographic algorithm that is intended to use the key.
- **Schemes or Modes of Operation:** This element defines the applicable schemes or modes of operation for performing a cryptographic function using a key.
- **Parameters for the Key:** This element specifies the parameters, if applicable, for a key.
- **Length of the Key:** This element specifies the length of the key in bits (or bytes).

Possible Metadata 3

- **Security Strength of the Key-Algorithm Pair:** This element is a number indicating the amount of work (that is, the base 2 logarithm of the number of operations) that is required to break (i.e., cryptanalyze) the cryptographic algorithm.
- **Key Type:** This element identifies the key type.
- **Appropriate Applications for the Key:** This element specifies applications for which the key may be used.
- **Security Policies Applicable to the Key Type:** This element identifies the security policies applicable to the key type.
- **Key Access Control List (ACL):** An access control list identifies the entities that can access and/or use the keys
- **Key Usage Count:** This element indicates the number of times that the key has been used for a specified purpose (e.g., encryption, decryption, sign, verify, wrap, or rekey).
- **Parent Key:** This element points to the key from which the key associated with this metadata is derived.

Possible Metadata 4

- **Key Sensitivity:** This element specifies the sensitivity or importance of the key. It could relate to a risk level (e.g., Low, Medium, or High) or a classification level (e.g., Confidential, Secret, or Top Secret)
- **Key Protections:** This element specifies the integrity, confidentiality, and source authentication protections applied to the key.
- **Metadata Protections** (can be a subset of the key protections or can be different)**:** This element specifies the mechanisms used to provide integrity, confidentiality, and source authentication to the associated metadata.
- **Trusted Association Protections:** how the trusted association of metadata to the key is protected

Metadata Date-Time Values 1

- **Generation date:** The date-time that a key was generated,
- **Association date:** The date-time that a key was associated with its metadata,
- **Activation date:** The date-time that a key was first used,
- **Future activation date:** The date-time that a key is to be used for the first time,
- **Renewal date:** The date-time that a public key was renewed and allowed to be used for a longer period of time, e.g., by generating a new certificate for the same public key as was provided in an old certificate
- **Future renewal data:** The date-time that a public key is to be renewed and allowed to be used for a longer period of time
- **Date of the last rekey:** The date-time that a key was replaced with a new key that was generated so that it is completely independent of the key that was replaced

Metadata Date-Time Values 2

- **Future rekey date:** The date-time that the key is to be replaced with a new key that will be generated so that it is completely independent of the key being replaced,
- **Date of the last key usage:** The date-time that the key was last used for a specified purpose,
- **Deactivation date:** The date-time that a key was deactivated,
- **Future deactivation date:** The date-time that a key is to be deactivated,
- **Expiration date:** The date-time that a key's useful lifetime was terminated permanently,
- **Revocation date:** The date-time after which a key was no longer considered valid,
- **Compromise date:** The date-time that a key a key was known to have been compromised and was marked for replacement and not renewal,
- **Destruction date:** The date-time that a key was destroyed, and
- **Future destruction date:** The date-time that a key is to be destroyed.

Key Management Functions 1

- Generate Key
  - The key generation methods and underlying random number generators.
- Register Owner
  - The initial registration of a security entity and a cryptographic key with metadata is a fundamental requirement of every CKMS. This requirement is difficult to fully automate while preserving security (i.e., protecting from the impersonation threat) and thus, it usually requires human interactions.

Key Management Functions 2

- Activate/Deactivate Key
  - Normal lifecycle events during processing
- Suspend/Reactivate Key
  - Temporary response to an unusual condition
- Revoke Key
  - A cryptographic key should be revoked as soon as feasible after it is no longer authorized for use (e.g., the key has been compromised).
- Key Renewal
  - Renewal establishes a new validity period for an existing subject public key beyond its previous validity period by issuing a new certificate containing the same public key with a new validity period

Key Management Functions 3

- Key Derivation
  - Obtaining a key from another secret value
- Key Update
  - Replacing a key with another derived from the first
- Key/Metadata Destruction
  - Burning paper, overwriting or destroying digital keys, including in backup storage media.

Section 7 – Interoperability

- Interoperability is achieved through the use of:
  - Common interfaces and protocols
  - Common formats for keys, metadata and other exchanged data
  - Compatible data exchange methods and security mechanisms between systems

Transitioning

- Switching from one algorithm or key length to another
  - a smooth transition often requires the capability to support the use of at least two algorithms or key lengths simultaneously
  - interoperability can be maintained until all participants have the capability to operate with the new algorithm or key length
  - the cryptographic protocols should be designed to identify and negotiate which algorithm and key length will be used in a particular key establishment transaction
- Current cryptographic algorithms should be implemented so that they can be augmented or replaced when needed.
  - See [SP 800-57-part1] and [SP 800-131A] for the NIST-recommended lifetimes of government-approved cryptographic algorithms.

Section 8 – Security Controls

- Physical Security
  - Fences, facilities, locks, alarms, cameras, visit logs, etc.
- Platform Security
  - Self-protection and isolation features
  - Access control
  - Event logging
  - Account management
- Malware Protection
- Auditing and Remote Monitoring
- Network Security Controls
- Cryptographic Module Controls

Section 9 – Testing and Assurance

- Vendor Testing
- Third-Party Testing
- Interoperability Testing
- Self-Tests
- Performance Tests
- Security and Functional Tests
- Limitations of Testing

Development Assurance

- Configuration Management
- Secure Delivery
- Development and Maintenance Environment Security
- Flaw Remediation Capabilities

Section 10 – Disaster Recovery

- Facility Damage
- Utility Service Outage
- Communication Outage
- System Hardware Failure
- System Software Failure
- Cryptographic Module Failure
- Key/metadata Corruption

Section 11 – Security Assessment

- Third-Party Validations
  - NIST’s Cryptographic Algorithm Validation Program (CAVP) tests correctness of algorithm implementations to their standards
  - NIST’s Cryptographic Module Validation Program (CMVP) validates cryptographic modules to FIPS 140-2 requirements
  - Common Criteria validation for non-cryptographic security components

Other Assessment Methods

- Architectural Review
- Functional and Security Testing
- Penetration Testing
- Periodic Security Review
- Security Maintenance

Section 12 – Technological Challenges

- New Attacks on Cryptographic Algorithms
  - A cryptographic algorithm has an expected security life. However, as time passes new attacks may be found that reduce its security life. This, in turn, is likely to reduce the security lifetime of the CKMS that relies on the algorithm to protect data.
- New Attacks on Key Establishment Protocols
  - Weaknesses are often found in protocols after they have been in use for several years.
- New Attacks on CKMS Devices
  - New methods for logically attacking computer-based systems are continuously being discovered.
- Quantum Computing
  - If large qubit quantum computers could be built, then the security of integer factorization and discrete log-based public-key cryptographic algorithms would be threatened.

Questions?