CRYPTREC GL-3004-1.1

暗号鍵管理ガイダンス

Part 1

独立行政法人情報処理推進機構

国立研究開発法人情報通信研究機構

目次

【修正履歴】

修正日	修正内容
2025.04.25（Ver.1.1）	名称を「暗号鍵管理ガイダンス Part 1」に変更
2023.05.09（Ver.1.0）	初版発行

1 はじめに

1.1 位置づけ

企業や個人の管理する情報を保護するために暗号アルゴリズムが広く利用されている。各暗号アルゴリズムは、それぞれの情報が必要とする機密性、完全性、認証を提供する目的で利用される。

デジタル庁と総務省、経済産業省は、暗号技術に関する有識者で構成される CRYPTREC 活動を通して、電子政府で利用される暗号技術の評価を行っており、2023 年 3 月に「電子政府における調達のために参照すべき暗号のリスト（CRYPTREC 暗号リスト）」を改定した。CRYPTREC 暗号リストは、安全性、実装性能及び市場における利用実績を踏まえ、「電子政府推奨暗号リスト」、「推奨候補暗号リスト」及び「運用監視暗号リスト」で構成される。

CRYPTREC 暗号リスト（電子政府推奨暗号リスト）：
https://www.cryptrec.go.jp/list.html

実際、「政府機関の情報セキュリティ対策のための統一基準（令和 3 年度版）[1]」（令和 3 年 7 月 7 日、サイバーセキュリティ戦略本部。以下、「統一基準」という）では、政府機関における情報システムの調達及び利用において、図 1-1 の通り、CRYPTREC 暗号リストのうち「電子政府推奨暗号リスト」に記載された暗号アルゴリズムを原則的に利用するように記載されている。このように、セキュアな暗号アルゴリズムの選択に関しては電子政府推奨暗号リストを活用する等により、比較的容易に満たすことができる。

しかしながら、実際のシステムがセキュアに動作し続けるためには暗号アルゴリズム自体がセキュアであるだけでは不十分である。統一基準でも暗号鍵の管理手順を定めることになっているように、データが保護される期間中、その暗号アルゴリズムが使用する暗号鍵もセキュアに管理されている必要がある。もし、暗号鍵がセキュアに管理されていなければ、管理が不十分な点を悪用した何らかの手段で暗号鍵が漏えいする可能性があり、その漏えいした暗号鍵を使ってシステムへの侵入、機密データの窃取や改ざん、なりすましなどが行われる。

一般に、暗号鍵管理の脆弱性を突く攻撃方法のほうが、セキュアな暗号アルゴリズム自体を解読するよりもはるかに容易な攻撃方法である。また、漏えいまでは至らなくても、暗号鍵にデータ不整合等が発生すればシステムエラーの原因となり、業務が停止するなどの悪影響が発生する場合もある。実際、セキュアな暗号アルゴリズムを利用していても、不十分な暗号鍵管理が原因となって、数多くのインシデントが発生している。

[1] 内閣サイバーセキュリティセンター（NISC）, https://www.nisc.go.jp/pdf/policy/general/kijyunr3.pdf

6.1.5 暗号・電子署名

遵守事項

(1) 暗号化機能・電子署名機能の導入

(a) 情報システムセキュリティ責任者は、情報システムで取り扱う情報の漏えいや改ざん等を防ぐため、以下の措置を講ずること。

（略）

(b) 情報システムセキュリティ責任者は、**暗号技術検討会及び関連委員会（CRYPTREC）により安全性及び実装性能が確認された「電子政府推奨暗号リスト」を参照**した上で、情報システムで使用する暗号及び電子署名のアルゴリズム並びにそれを利用した安全なプロトコル及びその運用方法について、以下の事項を含めて定めること。

(ア) 職員等が暗号化及び電子署名に対して使用するアルゴリズム及びそれを利用した安全なプロトコルについて、**「電子政府推奨暗号リスト」に記載された暗号化及び電子署名のアルゴリズムが使用可能な場合には、それを使用**させること。

(イ) 情報システムの新規構築又は更新に伴い、暗号化又は電子署名を導入する場合には、やむを得ない場合を除き、**「電子政府推奨暗号リスト」に記載されたアルゴリズム及びそれを利用した安全なプロトコルを採用**すること。

(ウ) 暗号化及び電子署名に使用するアルゴリズムが危殆化した場合又はそれを利用した安全なプロトコルに脆弱性が確認された場合を想定した緊急対応手順を定めること。

(エ) **暗号化された情報の復号又は電子署名の付与に用いる鍵について、管理手順を定める**こと。

（以下、略）

図 1-1　政府機関の情報セキュリティ対策のための統一基準（抜粋）

さらに、暗号鍵管理はうまく利用すると、大規模なデータ管理をセキュアに実現することも可能になる。例えば、クラウドサービスなど、外部の第三者にデータを預ける場合であっても、それらのデータを暗号化し、そのときの暗号鍵管理を利用者側が実施することで、クラウドサービス事業者に対しても機密性を維持できる。また、データセンタや大規模な記録メディアなどに保存されたデータで、物理的な破砕によるデータの完全削除を実現することが困難なケースでは、暗号鍵の破壊によって当該鍵で暗号化されたデータを事実上復号できなくすることでそれらのデータが完全に削除されたとみなす暗号化消去（Cryptographic Erase）といった方法を実現することもできる。

このような背景のもと、CRYPTREC では暗号鍵管理に関するガイドライン／ガイダンスを作成している。

- 暗号鍵管理システム設計指針（基本編）[2]
- 暗号鍵設定ガイダンス[3]
- 暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準[4]

このうち、「暗号鍵管理システム設計指針（基本編）」は、暗号鍵管理システム（以下、「CKMS (Cryptographic Key Management System)」という）を設計・構築・運用する際に参考すべきドキュメントとして作成されたものであり、「暗号鍵管理についての技術的内容」について解説している。具体的には、あらゆるユースケースにおける暗号鍵管理を安全に行うための構築・運用・役割・責任等に関する対応方針として考慮すべき事項一覧を提供し、CKMS 設計時に考慮すべきトピックス及び設計書等に明示的に記載する要求事項を示している。これは、CKMS の包括的な設計指針であり、CKMS 設計時に考慮すべきトピックス及び設計書等に明示的に記載する要求事項を列挙した NIST SP800-130「A Framework for Designing Cryptographic Key Management Systems」をベースに作成されている。

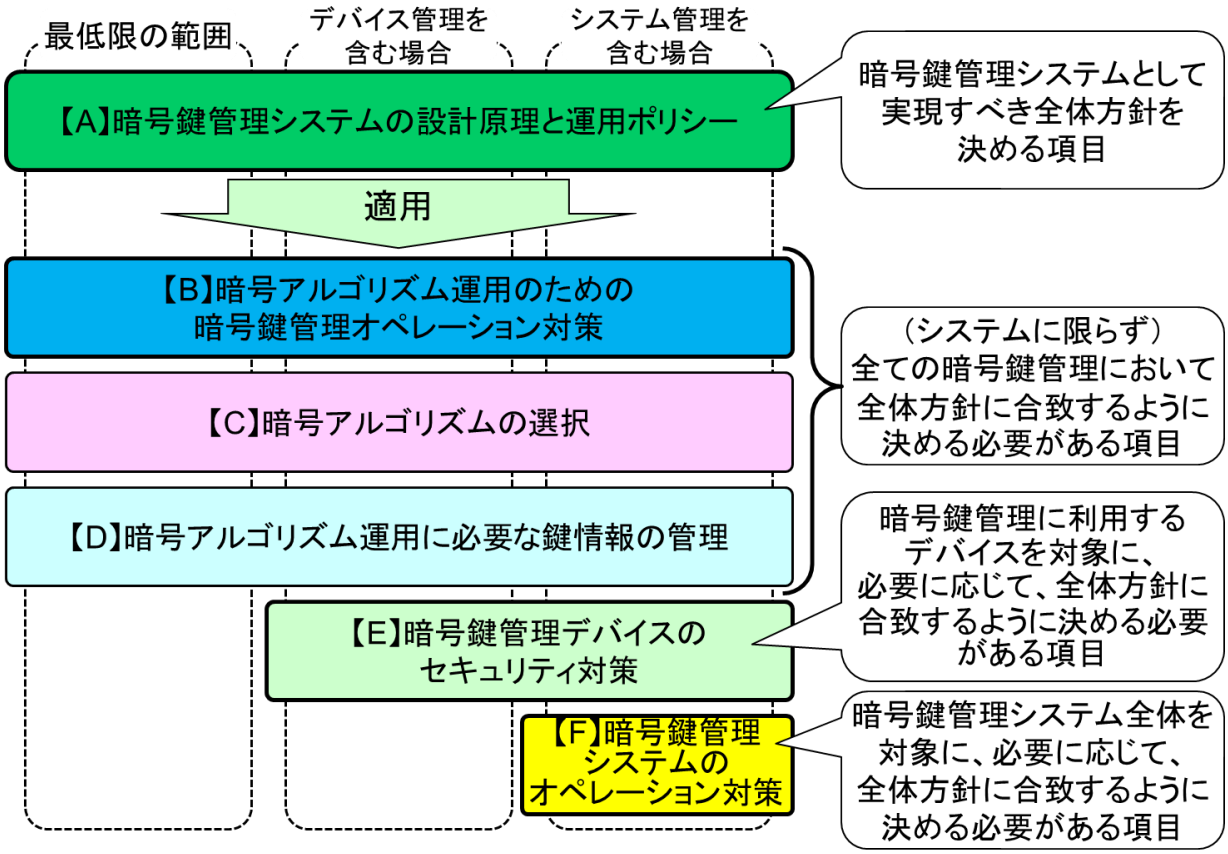


図 1-2　暗号鍵管理における目的別分類関係（「暗号鍵管理システム設計指針」より）

[2] https://www.cryptrec.go.jp/op_guidelines.html
[3] https://www.cryptrec.go.jp/op_guidelines.html
[4] https://www.cryptrec.go.jp/list.html

本ガイダンスは、本設計指針で記載が求められる項目について検討する際の有用な副読本となることを目的として書かれたものである。中でも、図 1-2 において、CKMS の利用環境に関わらず検討する必要がある項目のうちの【B】、【C】、【D】に該当する項目に関して、項目の概説及びその記載例を提供している。これらの項目は「狭義」の意味での暗号鍵管理に相当するものである。CKMS を設計する場合だけでなく、暗号アルゴリズムを使ったアプリケーション等を利用する場合なども含めて、全ての暗号鍵管理に対して検討が必要となる項目であることに留意されたい（ちなみに、【E】や【F】までを含む場合、「広義」の意味での暗号鍵管理を称す）。

1.2 想定読者

暗号鍵管理システム設計指針（基本編）での「暗号鍵管理についての技術的内容」での想定読者と同様、主として CKMS 設計者を想定読者としている。

1.3 構成

本設計指針は、4 章で構成されており、章立ては以下のとおりである。

2 章は「暗号アルゴリズム運用のための暗号鍵管理オペレーション対策」における項目についての解説・考慮点を示す。さらに、これらの理解を助けるため、簡単なシステム（トイモデル）を具体的に取り上げ、そのシステムで設定された構成や運用条件などを踏まえた場合の各々の項目における記載例を示す。

【トイモデルにおける注意】

ここでのトイモデルの構成や運用条件は、これらの内容と各々の項目における記載例との対応関係が“理解しやすくなる”ように設けたものであり、これらの内容を“推奨しているわけではない”ことに十分に注意されたい。

同様に、3 章では「暗号アルゴリズムの選択」における項目についての解説・考慮点を記載し、トイモデルでの記載例を示す。

4 章では、「暗号アルゴリズム運用に必要な鍵情報の管理」における項目についての解説・考慮点を記載し、トイモデルでの記載例を示す。

なお、本ガイダンスは、暗号鍵管理システム設計指針（基本編）と合わせて利用することを想定している。また、「暗号鍵管理システム設計指針（基本編）チェックリスト[5]」を利用する際に有用である。

[5] IPA, https://www.ipa.go.jp/security/crypto/guideline/ckms.html

1.4　検討体制

本ガイダンスは、2021 年度及び 2022 年度 CRYPTREC 暗号鍵管理ガイダンス WG において作成された。

表 1-1　暗号鍵管理ガイダンス WG の構成（2023 年 3 月時点）

主査	上原　哲太郎	立命館大学　情報理工学部　情報理工学科　教授
委員	漆嶌　賢二	GMO グローバルサイン株式会社　プロダクトマネジメント部 部長
委員	垣内　由梨香	Microsoft Corporation　セキュリティレスポンスチーム セキュリティプログラムマネージャー
委員	菅野　哲	GMO サイバーセキュリティ by イエラエ株式会社 取締役 CTO of Development
委員	菊池　浩明	明治大学　総合数理学部　先端メディアサイエンス学科　教授
委員	小林　浩二	パナソニックオートモーティブシステムズ株式会社 開発本部　プラットフォーム開発センター セキュリティ開発部セキュリティ PF 開発課　係長
委員	須賀　祐治	株式会社インターネットイニシアティブ セキュリティ本部セキュリティ情報統括室　シニアエンジニア
委員	西原　敏夫	シスコシステムズ合同会社　カスタマーエクスペリエンス シニアセキュリティアーキテクト
委員	舟木　康浩	タレス DIS CPL ジャパン株式会社 クラウドプロテクション&ライセンシング データプロテクション事業本部　セールスエンジニアマネージャ
委員	満塩　尚史	デジタル庁　戦略・組織グループ　セキュリティ危機管理チーム セキュリティアーキテクト

2 暗号アルゴリズム運用のための暗号鍵管理オペレーション対策

本章の目的・趣旨

本章は、設計指針（基本編）の 5 章に記載されている要求事項（各節での色付き枠内で示している内容）について解説したものである。

CKMS においてどのように暗号鍵が管理されるかを記載するものであり、「狭義」の意味での暗号鍵管理に相当するものである。暗号鍵の生成から廃棄までのライフサイクル全期間にわたって暗号鍵を管理するのに必要となる機能や運用方法を取り決める項目（B.01～B.81）を集めている。ここには、主に以下のような項目を含んでいる。

- 各暗号鍵は、どのような処理を目的として使われるのか。
- それら暗号鍵の保管場所や保管方法
- 各暗号鍵の生成から廃棄までのライフサイクルを通し、暗号鍵がどのように管理されるか。またその管理を行う上で必要となる機能群はどのようなものか。

CKMS を設計する場合だけでなく、暗号アルゴリズムを使ったアプリケーション等を利用する場合なども含めて、全ての暗号鍵管理に対して検討が必要となる項目であることに留意されたい。

2.1 CKMS 設計

① 暗号鍵を提供するために CKMS をどのように構築するかの概要

項目	FR 番号	Framework Requirements の内容	SP800-130
B.01	FR2.4	CKMS 設計は、以下を含む CKMS システムの高レベルの概要を明記しなければならない： a) 利用するそれぞれの鍵タイプ b) 鍵が生成される場所と手段 c) それぞれの鍵タイプとの信頼関係で使用されるメタデータ要素（7.1 節表 7 2 参照） d) 鍵情報（暗号鍵やメタデータ）が存在しているそれぞれのエンティティのストレージにおける、鍵情報（暗号鍵やメタデータ）の保護方法 e) 配送時の鍵情報（暗号鍵やメタデータ）の保護方法 f) 鍵情報（暗号鍵やメタデータ）が配送され得る先となるエンティティの種類（例えば、ユーザ、ユーザデバイス、ネットワークデバイス）	2.5 節

解説・考慮点

項目 B.01 は、CKMS の設計にあたって、暗号鍵を提供するために CKMS をどのように構築するかの概要を明確化することを要求したものである。ここでは、機微なデータを保護するための暗号鍵に対する設計方針や実現目標、詳細を決める文書へのインデックス等を簡潔な概要で明記することが求められる。

本節で要求していることは、CKMS をどのような設計方針の下でどのように構築されるのかの高レベルの概要を整理し、明らかにしておくことである。ここでの「高レベルの概要」の意味は、次節以降に決める必要がある事項を検討する際に本概要で定めたことと矛盾していないことが確認できる程度に具体化した情報、ということである。

CKMS 設定の詳細については次節以降で取り扱うことになるので、ここであまり詳細に定める必要はない。

重要なことは、安全な CKMS 設計の第一歩として、B.01 において少なくとも以下のような事項を検討し、明らかにしておくことで、次節以降で詳細を定めなければいけない項目に抜けが生じないようにすることである。また、ここでの概要は CKMS の基本設定となるものであるので、次節以降では、本概要に記載したことに矛盾するような内容を定めてはならないことに注意する必要がある。

- a）と c）では、「どのような種類」の暗号鍵（及びそれに付随するメタデータ）を保護管理対象とする必要があるのかの把握
- b）では、暗号鍵が「どのように生成」されるのかの概要把握
- d）と e）では、暗号鍵が「どのような場所でどのように保護」されるのかの概要把握
- f）では、暗号鍵を「誰」が所持したり利用したりするのかの特定

その際、注意する必要があるのは、CKMS 設計としてどこまでの範囲を対象とするのかであり、それに応じて境界を定めることである。この範囲は、CKMS が取り扱う目的や設置場所、利用するエンティティやアプリケーション、プロトコルなどにより異なる。

例えば、図 2-1 のようなシステム（System A、System B、System C）間で、暗号鍵管理を含む暗号処理全体を担う各々の CKMS モジュールを経由して暗号通信をするプロトコルを利用するシステムを対象とした場合、CKMS 設計の対象範囲はシステム内での全ての CKMS モジュール（図 2-1 では CKMS Module A、CKMS Module B、CKMS Module C）の和集合、及び個々の CKMS モジュールと連携して動作する機器類やそのモジュールを利用するエンティティなどから構成される。一方、System A 内部に閉じたデータ保管を対象とした場合には、CKMS 設計の対象範囲は System A の CKMS モジュール（図 2-1 では CKMS Module A）、及び当該 CKMS モジュールと連携して動作する機器類やそのモジュールを利用するエンティティなどだけとなる。

この範囲を正しく定めておかなければ、本来管理対象とすべき暗号鍵が対象から漏れ、必要な対策が取られなかったり、逆に、不必要に多くの暗号鍵が管理対象となり、無駄な対策を実施することで余分なコストがかかったり利便性が低下したり、といった問題が生じることとなることに留意されたい。

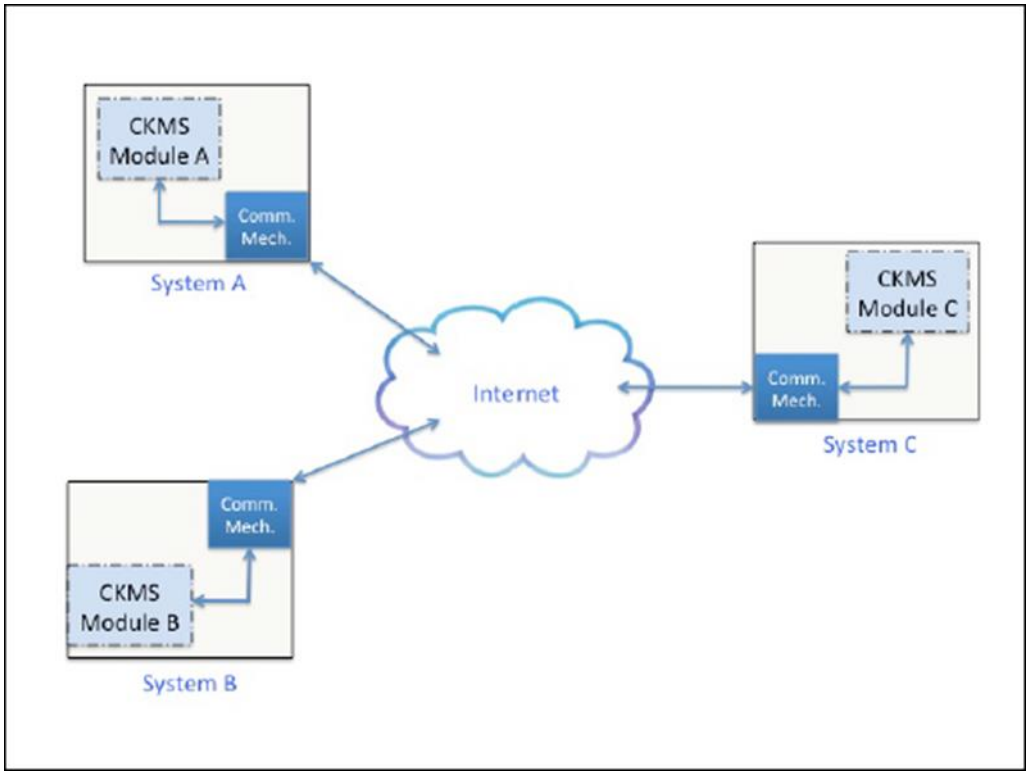


図 2-1　CKMS 概要例（NIST SP 800-130 より）

《トイモデルと記載例》

本節のトイモデルは、図 2-2～図 2-4 に示す、メールの送信元認証を S/MIME の署名付きメールで実現するシステムとする。

このシステムの CKMS 設計範囲は、図 2-2 の通り、メールの署名生成と署名検証の処理、署名生成や署名検証に使う暗号鍵の管理、公開鍵証明書の申請処理である。メールの送受信などを処理する通信モジュールは含まない。また、CA は外部のパブリック CA であり、これも CKMS には含まない。

準備として、図 2-3 の通り、メール送信者は、メール管理部において、信頼できる手段で署名プライベート鍵と署名公開鍵のペアを作成する。次いで、その署名公開鍵の CSR（証明書署名要求; Certificate Signing Request）を作成し、その CSR を組織内の情報システム部に送る。情報システム部は、証明書管理部において、CSR を確認し、確認後に CSR をパブリック CA に送り、CA が署名した公開鍵証明書を情報システム部が受信し、確認後にその公開鍵証明書をメール送信者に送信する。なお、CKMS の対象範囲はメール送受信 PC のメール管理部と情報システム部の証明書管理部だけであるので、図 2-3 での色が塗られた部分が該当する。

メールの署名と検証は図 2-4 の通りであり、メール送信者は、メール送信 PC のメール管理部において、送信するドキュメントのハッシュ値を計算し、そのハッシュ値に自分の署名プライベート鍵を使って署名し、ドキュメント、署名、自分の署名公開鍵の公開鍵証明書をメールで送信する。

メール受信者は、メール受信 PC のメール管理部において、(a)メール送信者の公開鍵証明書が有効であることを CA 証明書を使って確認し、(b)受信したメールからドキュメントを取り出し、ハッシュ値を計算し、その値とメール送信者の公開鍵証明書から取り出した公開鍵で署名が正しいか検証し、メール送信者から送られたメールであることを確認する。

なお、CKMS の対象範囲はメール送受信 PC のメール管理部となるので、図 2-4 での色が塗られた部分が該当する。また、パブリック CA は CKMS の対象範囲外としているので、CA が使う署名プライベート鍵や対応する署名公開鍵は CA が適正に管理していることを前提とし、その前提に問題がないことだけを CA 証明書によって確認する。

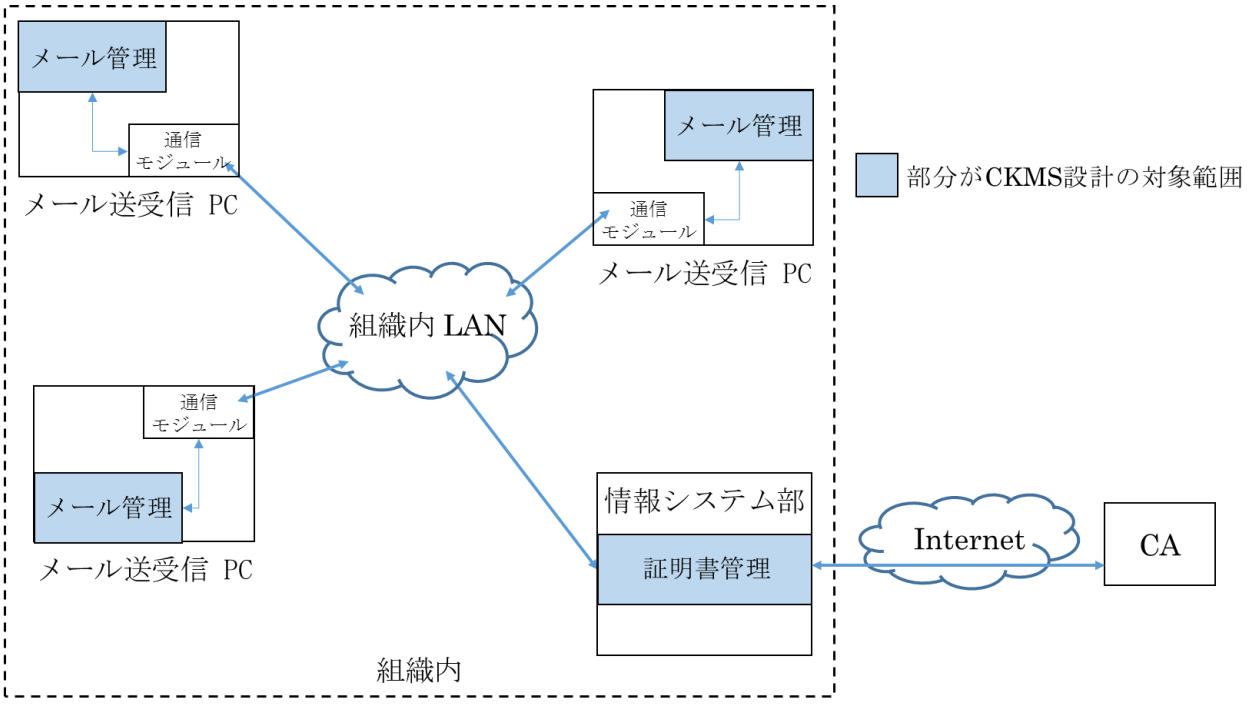


図 2-2　トイモデルでの CKMS 概要図

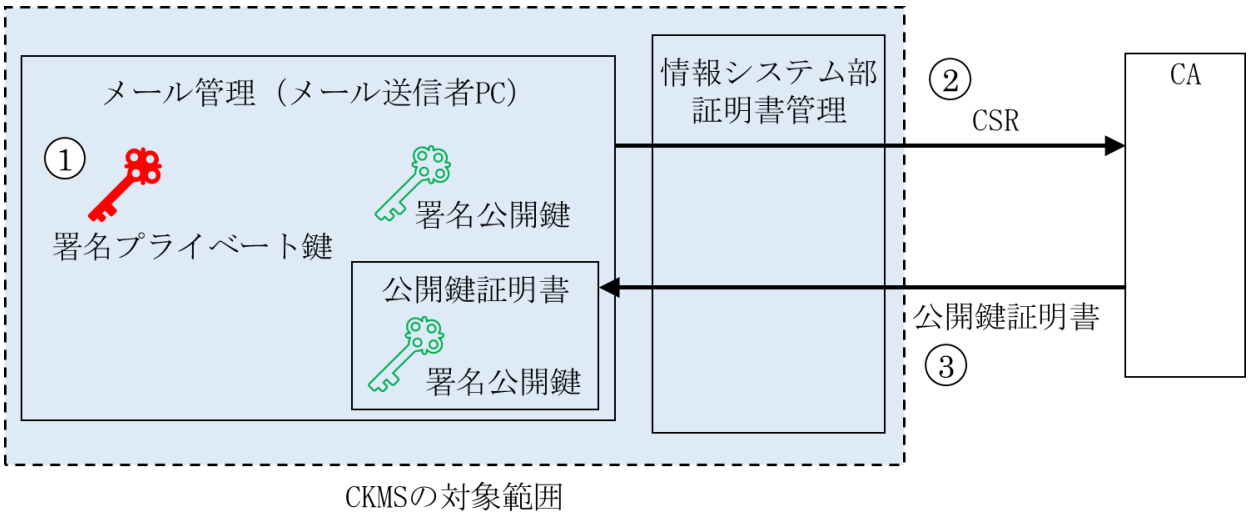


図 2-3　鍵生成、CSR 送信、証明書受信

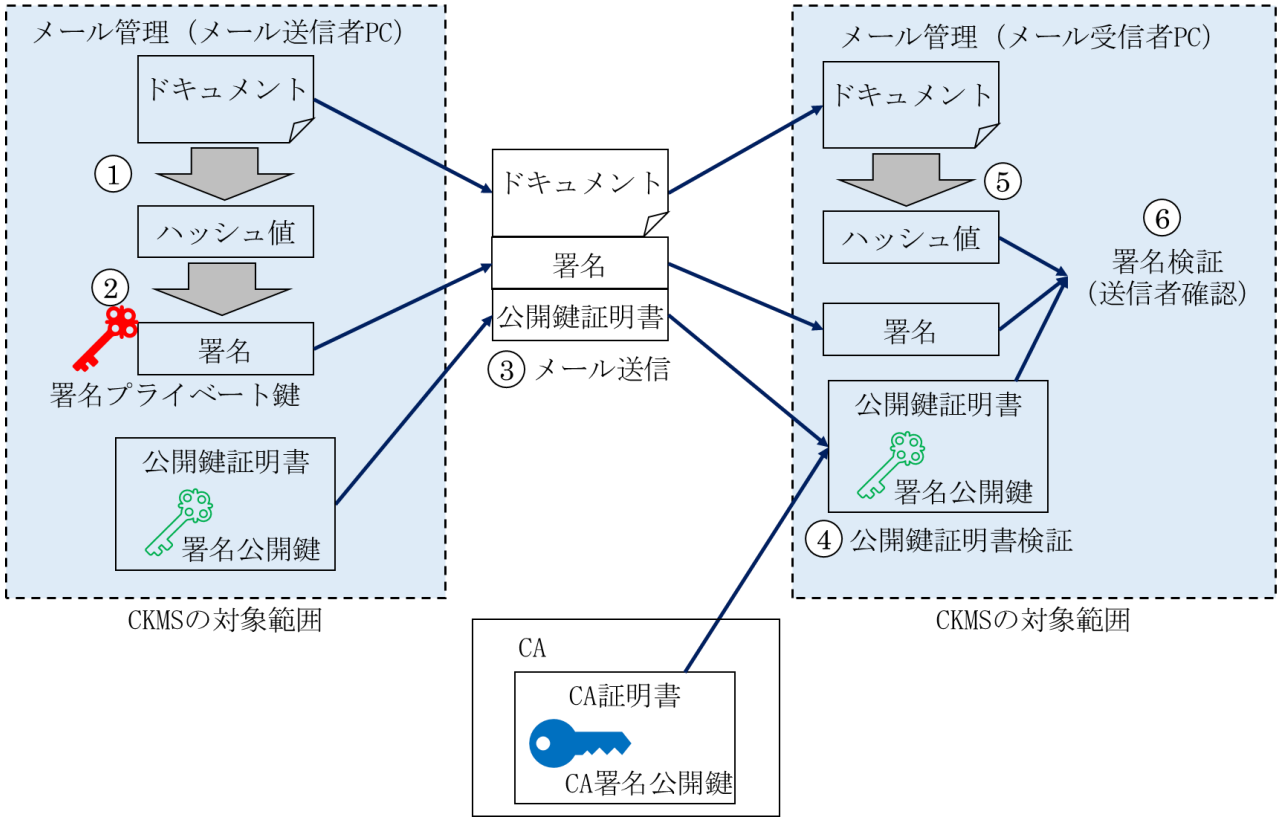


図 2-4　メールの署名と検証

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.01	a)　利用するそれぞれの鍵タイプ 署名プライベート鍵、署名公開鍵 b)　鍵が生成される場所と手段 鍵生成はメール送信者の PC で行われる c)　それぞれの鍵タイプとの信頼関係で使用されるメタデータ要素（4.1 節参照） 鍵の有効期間、親鍵（CA の署名公開鍵）、CA との信頼関係 d)　鍵情報（暗号鍵やメタデータ）が存在しているそれぞれのエンティティのストレージにおける、鍵情報（暗号鍵やメタデータ）の保護方法 署名プライベート鍵はアクセスコントロールで保護される e)　配送時の鍵情報（暗号鍵やメタデータ）の保護方法 メール受信者に送信する公開鍵証明書は CA が署名している f)　鍵情報（暗号鍵やメタデータ）が配送され得る先となるエンティティの種類（例えば、ユーザ、ユーザデバイス、ネットワークデバイス） 公開鍵証明書はメールを送受信するユーザに配布

2.2　暗号鍵のライフサイクル

①　暗号鍵のライフサイクル全体にわたって取り得る鍵状態及び遷移条件の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.02	FR6.15	CKMS 設計は、CKMS の鍵が取り得る全ての状態を明記しなければならない。	6.3 節
B.03	FR6.16	CKMS 設計は、全ての CKMS 鍵状態間の遷移、及び遷移を起こすことに関係するデータ（入力と出力）を明記しなければならない。	6.3 節

解説・考慮点

暗号鍵のライフサイクルの一般形は、SP 800-57 part1 の 7 節「鍵状態と遷移（Key States and Transitions）」に基づく。これをベースに、CKMS とそのアプリケーションに適切な鍵状態と遷移条件を選択し定義する。
項目 B.02 及び B.03 は、CKMS の設計にあたって、暗号鍵のライフサイクル全体を対象に定義した全ての鍵状態及び遷移条件を明確化することを要求したものである。ここで定義した鍵状態及び遷移条件を管理・実行するために必要な全ての管理機能を次節以降で規定することが求められる。

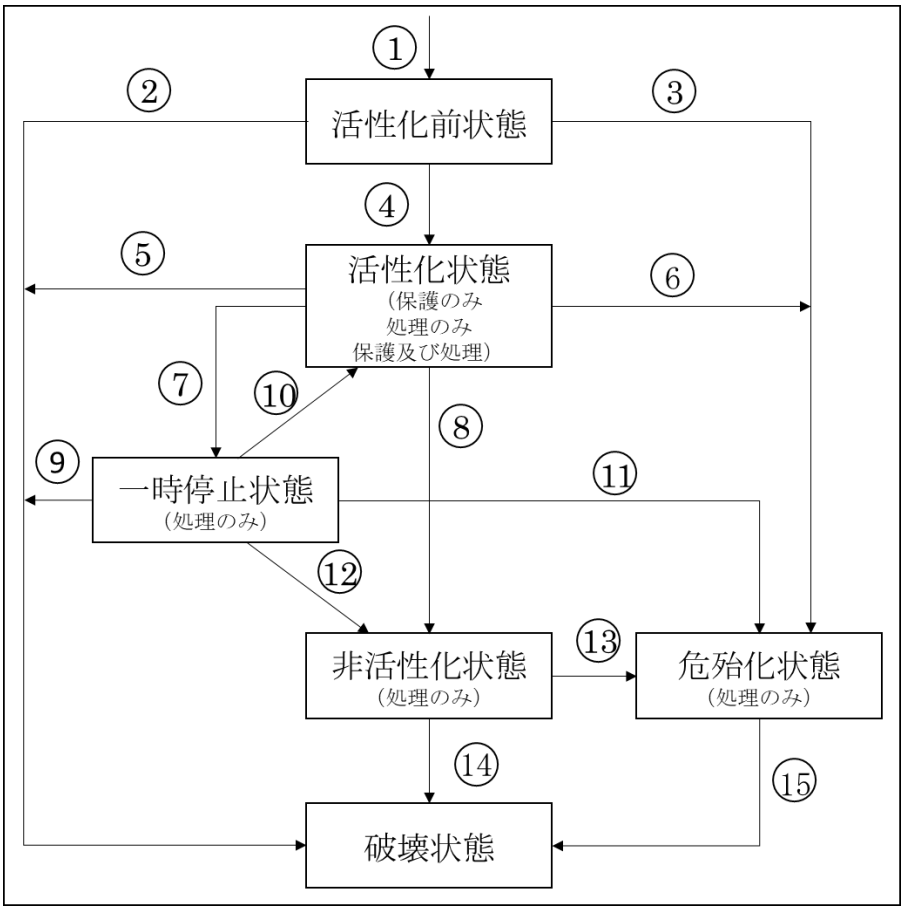
暗号鍵は、生成から破壊までの間（ライフサイクルという）にいくつかの状態を経由することとなる。これらの状態に応じて暗号鍵は異なる方法で使用される（又は使用が禁止される）ので、CKMS 設計の観点からは、どのような状態が存在し、どのような条件によって状態遷移が起きるのかを明らかにしておく必要がある。

項目 B.02 及び B.03 は、CKMS の設計にあたって、そのような暗号鍵のライフサイクル全体を対象に全ての鍵状態及び遷移条件を明確化することを要求したものである。その際、B.02 では、項目 B.01 で記載した暗号鍵が遷移する必要がある全ての状態を含まなければならない。

また、遷移の処理は次節でのライフサイクル管理機能により実現され、遷移条件に基づいて暗号鍵の状態がコントロールされるようにしなければならないので、B.02 及び B.03 で記載したことと次節での内容とが整合的であるようにしなければならないことに注意する必要がある。

暗号鍵の状態及び遷移の詳細については、NIST SP800-57 Part1 Rev5[6]の第 7 章を参照されたい。図 2-5 にその概要を記す。

[6] https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final
日本語訳：https://www.ipa.go.jp/security/crypto/archive.html



【遷移】

- ✧ 正常な遷移：①→④→⑧→⑭
- ✧ それ以外はすべて何らかの異常による遷移

図 2-5　鍵状態及び遷移の例

- 活性化前状態：

  暗号鍵は生成されたが、使用が認可されていない状態のこと。鍵の所持証明や確認など、暗号鍵の検証にのみ使用することができ、それ以外の目的（例：暗号化や署名などの処理）に使ってはいけない。

  また、活性化前状態に遷移する段階（図 2-5 の①）で、暗号鍵の生成やエンティティ・所有者の確認などが行われる。

- 活性化状態：

  暗号鍵が、実際の暗号処理で使用することを認可されている状態のこと。暗号鍵のタイプに応じて、保護のみ（暗号化や署名生成）、処理のみ（復号や署名検証）、又は保護と処理の両方のいずれかを選んで指定できる。

  また、活性化前状態からの遷移は（図 2-5 の④）は、活性化前状態に入った直後に遷移する場合もあれば、何らかのトリガー（開始日時など）により遷移する場合もある。

- 非活性化状態：
  暗号鍵が、保護（暗号化や署名生成）を行うために使用してはならないが、場合によっては、暗号化された保護情報を処理（復号や署名検証）するために使用されうる状態のこと。例えば、アーカイブされている暗号鍵が非活性化状態に該当していることがあり得る。また、活性化状態は終了したが破壊状態になる前の暗号鍵も非活性化状態として取り扱われる。
  活性化状態からの遷移は（図 2-5 の⑧）は、基本的に当該鍵の有効期限が終了したタイミングで行われる。

- 破壊状態[7]：
  暗号鍵が完全に存在しなくなった状態のこと。バックアップやアーカイブされている暗号鍵についても（もしあれば同時に）破壊することが求められる。
  暗号鍵が破壊されると、それ以降、当該暗号鍵の危殆化を心配する必要はなくなる。一方で、当該暗号鍵で保護（暗号化）を行った情報を処理（復号）することはできなくなり、当該情報を永遠に喪失する結果を招くことになるので、誤って暗号鍵の破壊をしてしまうことがないようにしておくことも重要である。

- 危殆化状態：
  暗号鍵が、実際の暗号処理で安全に利用できる保証がなくなった状態、又はその可能性が生じた状態のこと。例えば、暗号鍵の漏えいや暴露、認可されていないエンティティによる不正アクセスなどが生じた時（又はその疑いが生じた時）に危殆化状態への遷移（図 2-5 の③⑥⑪⑬）を実施する。
  危殆化状態となった暗号鍵（及びその鍵ペア）は失効させる必要があり、その後、保護（暗号化や署名生成）を行うために使用されてはならない。また、暗号化された保護情報を処理（復号や署名検証）する場合には、その情報の正当性や完全性が疑わしくなっていることを十分に認識したうえで、どのように取り扱うのかを考える必要がある。例えば、危殆化の発生前から物理的に保護されているか、信頼できるタイムスタンプの利用など別の保護手段により保護されているか、などの視点を踏まえて、当該情報の正当性や完全性を受け入れるか受け入れないかを判断することが必要となる。

- 一時停止状態：
  暗号鍵の利用が一時的に認められなくなっている状態のこと。一時停止状態にある暗号鍵は、保護（暗号化や署名生成）を行うために使用されてはならない。
  一般に、一時停止状態への遷移（図 2-5 の⑦）には、i）暗号鍵の危殆化が疑われ、その状況を調査するための時間を確保しつつ、万が一の危殆化の影響が発生しないようにするケースと、ii）当該暗号鍵を所有するエンティティが一定期間利用しない（長期休暇など）こ

[7] 設計指針（基本編）では「破壊状態」、SP 800-57 の日本語訳では「破棄状態」と記載されているが、本ガイダンスでは「破壊状態」で統一する。「暗号鍵を破棄」といった場合、「（再利用できないように）暗号鍵の存在自体を完全に消す」場合と「（簡単には使えないように）暗号鍵を読み出せなくする」場合が考える。破壊状態とみなすのは前者だけであることに留意されたい。後者は、簡単には暗号鍵を読み出せなくなっているというだけでデータとしては残ったままの可能性があり、このような状態は破壊状態とはみなさない。

とが明らかであり予防保全を行うケース、のどちらかの理由に起因して実施されることが考えられる。i）の場合、調査の結果により、危殆化の恐れが払拭されれば活性化状態に遷移（図 2-5 の⑩）することもあり得るが、そうでなければ危殆化状態に遷移（図 2-5 の⑪）するのが普通である。ii）の場合は、利用しない期間が明ければ（休み明けにより利用再開など）即座に活性化状態に遷移（図 2-5 の⑩）するのが一般的である。

《トイモデルと記載例》

本節のトイモデルも 2.1 節と同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムとする。

図 2-6 の通り、署名プライベート鍵は署名生成（保護）のために利用するため、有効期間前の活性化前状態、有効期間内の活性化状態、有効期間後の非活性化状態に遷移する。その後、実際に署名プライベート鍵が破壊されると破壊状態に遷移する。また、署名プライベート鍵の危殆化が疑われる場合は危殆化状態に遷移させる。なお、本システムでは一時停止状態は設定しない。

署名プライベート鍵のペアとなる署名公開鍵についても、有効期間前の活性化前状態、有効期間内の活性化状態、有効期間後の非活性化状態に遷移する。その後、実際に署名プライベート鍵が破壊されると破壊状態に遷移する。また、署名プライベート鍵が危殆化状態に遷移した場合、署名公開鍵も同時に危殆化状態に遷移させる。これら以外の遷移条件は設定しないこととする。

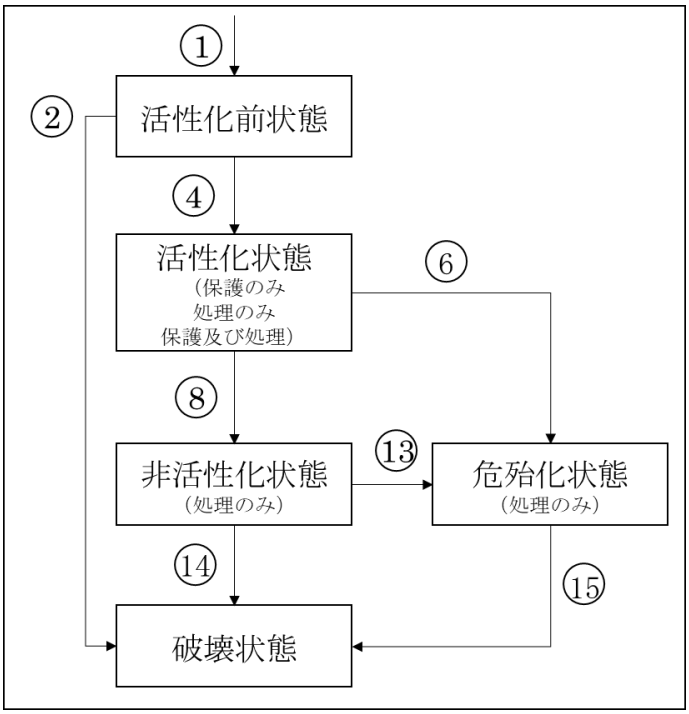


図 2-6　トイモデルの鍵状態及び遷移

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.02	活性化前状態、活性化状態、危殆化状態、非活性化状態、破壊状態
B.03	● 署名プライベート鍵と署名公開鍵の鍵ペア生成後に活性化前状態に遷移（図 2-6 の①） ● 証明書に記載された有効期間の開始時に活性化前状態から活性化状態に遷移（図 2-6 の④） ● 活性化状態に遷移前に署名プライベート鍵に問題が生じた場合は、活性化前状態から破壊状態へ遷移（図 2-6 の②） ● 有効期限の終了時に活性化状態から非活性化状態に遷移（図 2-6 の⑧） ● 鍵の破壊条件を満たした場合に非活性化状態から破壊状態に遷移（図 2-6 の⑭） ● 活性化状態に遷移後に署名プライベート鍵の危殆化が疑われる事象が発生した場合は、活性化状態又は非活性化状態から危殆化状態に遷移（図 2-6 の⑥⑬）。署名公開鍵も同時に危殆化状態に遷移させる ● 証明書失効リスト（CRL リスト）に記載された署名公開鍵は危殆化状態に遷移（図 2-6 の⑥⑬） ● 危殆化処理完了時に破壊状態に遷移（図 2-6 の⑮） 注）本システムでは、図 2-5 の③⑤⑦⑨⑩⑪⑫の遷移は設定しない。

2.3　暗号鍵のライフサイクル管理機能

①　鍵情報に対する管理のために実行される機能の全体像

項目	FR 番号	Framework Requirements の内容	SP800-130
B.04	FR6.17	CKMS 設計は、実装されサポートされる鍵情報（暗号鍵及びメタデータ）の管理機能を明記しなければならない。	6.4 節
B.05	FR6.18	CKMS 設計は、CKMS に実装されるそれぞれの鍵情報（暗号鍵及びメタデータ）の管理機能のパラメタに適用される完全性、機密性、及びソース認証（source-authentication）の処理（service）を特定しなければならない。	6.4 節

解説・考慮点

項目 B.04 は、CKMS の設計にあたって定義した暗号鍵のライフサイクルにおける鍵状態及び遷移条件を管理・実行するために必要な全ての管理機能を明確化し、実装することを要求したものである。対象となる管理機能は本節の②以降である。
項目 B.05 は、管理機能に共通して入出力されるデータについて、完全性や機密性、ソース認証が必要となるものがあれば、それらを明確化することを要求したものである。これには、エンティティの認証及び認可が含まれることもある。

項目 B.04 の目的は、暗号鍵のライフサイクル管理機能として②以降で対象となる機能を全て明記することによって、詳細を定めなければいけない項目に抜けが生じないようにすることである。なお、それぞれの機能の詳細については次節以降で取り扱うことになるのでここであまり詳細に定める必要はないが、前節にも記載した通り、ライフサイクル管理機能が暗号鍵のライフサイクルを実現するための手段であるので、その基本方針となる B.02 及び B.03 で記載したことと整合的であるようにしなければならない。

項目 B.05 の目的は、管理機能に入出力される個々の鍵情報（暗号鍵及びメタデータ）が、完全性や機密性、ソース認証のどの処理（service）に使われるものなのかを明記することにより、鍵情報が誤った使い方をされていないことを確認することである。したがって、B.01 で記載した暗号鍵全てを包含していることが求められる。また、②以降での処理（service）についての項目において、B.05 と矛盾していないことを確認する必要がある。

②　鍵活性化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.06	FR6.22	CKMS 設計は、それぞれの鍵タイプがどのように活性化されるか、及び鍵が活性化される状況を明記しなければならない。	6.4.3 節
B.07	FR6.23	それぞれの鍵タイプに対して、CKMS 設計は、鍵活性化の通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.3 節

解説・考慮点

暗号鍵の活性化前状態から活性化状態への遷移を提供する機能である。
項目 B.06 及び B.07 は、CKMS の設計にあたって、鍵活性化の手順や遷移条件、通知方法等、鍵活性化機能への要求事項を明確化することを求めたものである。

項目 B.06 は、活性化前状態から活性化状態への遷移（図 2-5 の④）を実現するための手順や遷移条件を具体化することを求めたものであり、B.03 で記載した活性化状態への遷移条件と整合的であるようにしなければならない。

項目 B.07 は、暗号鍵が活性化状態になった時に、当該暗号鍵を利用するエンティティ（利用者）に、その暗号鍵が暗号処理に利用できるようになったことを通知する方法を具体化することを求めたものである。例えば、データ暗号化対称鍵であれば当該鍵の利用エンティティだけに秘密裏に通知できる手段を使って通知する必要がある一方、署名公開鍵であれば多くのエンティティに周知できる手段を使う必要があるかもしれない。

活性化状態に遷移したことが通知されない場合、当該暗号鍵を利用するエンティティ（利用者）はいつからその暗号鍵が使えるのかが分からないため、基本的には活性化状態へ遷移する前にい

つから使えるのか通知することが必要であると考えるのがよい。ただし、通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

③　暗号機能の実行場所の特定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.08	FR6.69	CKMS 設計は、サポートされている全ての暗号機能、及びそれらの暗号機能が CKMS のどこで実行されるか（例えば、CA、ホスト、又はエンドユーザシステム）を明記しなければならない。	6.4.27 節

解説・考慮点

データへの暗号学的保護を実際に提供する機能であり、署名生成、署名検証、暗号化、復号、鍵ラッピング、鍵アンラッピング、MAC 生成、及び MAC 検証を含む。
項目 B.08 は、CKMS の設計にあたって、暗号機能がどこで実行されるのか明確化することを要求したものである。

項目 B.08 は、暗号機能がどこに存在し、どこで実行されるのかを把握しておくことを求めたものである。暗号機能が実行されるところでは必ず暗号鍵が平文の形で使われることになるため、暗号鍵が保管される場所と並んで、もっとも暗号鍵の危殆化が発生しやすい場所である。したがって、暗号機能がある場所や実行場所を把握しておくことで、暗号鍵が狙われるリスクを低減させるために重点的に対策・保護すべき場所の絞り込みに活用できる。

例えば、暗号モジュールの内部で暗号機能が実行されることとなれば、暗号鍵が平文の形で使われるのは暗号モジュール内に限定される。

④　鍵非活性化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.09	FR6.24	CKMS 設計は、各鍵タイプに対して、鍵の非活性化がどのように決定されるのか（例えば、暗号鍵有効期間（cryptoperiod）による、使用回数による、又はデータ量による）を明記しなければならない。	6.4.4 節
B.10	FR6.25	CKMS 設計は、それぞれの鍵タイプがどのように非活性化されるか（例えば、非活性化日時、使用回数、又は保護されたデータの量に基づいて、手動で行われるのか自動で行われるのか）を明記しなければならない。	6.4.4 節

B.11	FR6.26	CKMS 設計は、それぞれの鍵タイプの非活性化日時がどのように変更できるかを明記しなければならない。	6.4.4 節
B.12	FR6.27	それぞれの鍵タイプに対して、CKMS 設計は、鍵タイプの非活性化の事前通知の要求事項を明記しなければならない。それには、CKMS がサポートするどの役割に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.4 節

解説・考慮点

暗号鍵の非活性化状態への遷移を提供する機能である。CKMS セキュリティポリシーには当該ポリシーがカバーするあらゆる鍵タイプについて最大許容暗号鍵有効期間を記載すべきであり、その期間を超えた暗号鍵有効期間を設定してはならない。
項目 B.09～B.12 は、CKMS の設計にあたって、鍵非活性化の手順や遷移条件、変更方法、通知方法等、鍵非活性化機能への要求事項を明確化することを求めたものである。

セキュリティ対策の一環として、多くの場合、暗号鍵が無期限に使われることはなく、あるタイミングで暗号鍵の交換を行う。暗号鍵の活性化状態に存在している時間のことを暗号鍵有効期間という。この時間は、扱う情報の資産価値、求められる情報の機密性や完全性、CKMS への脅威、暗号鍵の交換に伴うメリットとデメリットの比較などに基づいて決められる。

暗号鍵有効期間が経過した暗号鍵は非活性化状態に遷移することで、新たな保護（暗号化や署名生成）を行うことはできなくなる。

項目 B.09 と B.10 は、管理対象となる各々の暗号鍵がどのような条件を満たしたときにどのような方法で非活性化状態に遷移するのかを決めるためのものである。例えば、遷移条件では、暗号鍵有効期間のほか、使用回数やデータ量などを使うケースもある。いずれの条件を使うとしても、CKMS セキュリティポリシーや CKMS 設計などで最大許容暗号鍵有効期間が決まっている場合には、その最大期間と矛盾しないようにしなければならない。また、遷移方法では、暗号鍵が自動的に利用できなくなるように非活性化状態に直接遷移する（図 2-5 の⑧）ような場合もあれば、例えば、一時停止状態に自動的に遷移（図 2-5 の⑦）した後、管理者等の確認処理を経て非活性化状態に遷移する（図 2-5 の⑫）ようなやり方も想定される。

項目 B.11 は、非活性化状態への遷移条件の変更を例外的に認めるかどうか、また、認める場合でもどのような条件下で認めるのかを決めておくものである。基本的には例外を認めない方がよいと考えられるが、システム上の要請などにより例外的に遷移条件の変更を容認する必要がある場合にはそのことを明確にしておくこと、また例外がなし崩しにならないような明確な条件として定めておくことが重要である。

なお、これらは B.03 の非活性化状態への遷移条件と整合していなければならない。

項目 B.12 は、暗号鍵が非活性化状態になるより前に、当該暗号鍵を利用するエンティティ（利用者）に、その暗号鍵が暗号処理に利用できなくなることを予告する方法を具体化することを求

めたものである。非活性化状態に遷移することが事前に通知されることなく、非活性化状態に遷移した場合、当該暗号鍵を利用するエンティティ（利用者）にとっては突然その暗号鍵が使えなくなることを意味し、それが故障などの予期せぬ原因によって生じた事態だと誤認する恐れがある。このような事態を避けるのが事前通知の役割であり、何らかの方法で通知する仕組みを設けたほうがよい。ただし、通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

⑤　鍵失効機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.13	FR6.28	CKMS 設計は、いつ、どのように、どのような状況で失効が実行され、失効情報を依拠する当事者が利用可能になるかを明記しなければならない。	6.4.5 節
B.14	FR6.108	CKMS 設計は、使用される又は使用できる鍵失効メカニズム及び関連付けられた依拠するエンティティへの通知メカニズムを明記しなければならない。	6.8.3 節

解説・考慮点

暗号鍵有効期間より前に当該暗号鍵の使用を終了させる必要が生じた場合に行われ、暗号鍵の危殆化状態への遷移を提供する機能である。この機能が実行されると、過去に保護された情報の処理のための使用に対しても完全なセキュリティは保証されないので、当該暗号鍵を速やかに置き換える能力とその鍵を使用する当事者に危殆化／失効を通知する能力を備えているべきである。
CKMS の設計にあたって、項目 B.13 は、鍵失効の遷移条件及び通知方法といった鍵失効機能への要求事項を明確化することを求めたものである。B.14 は、実際に利用する具体的な鍵失効メカニズム及び通知メカニズムを明確化することを要求したものである。

非活性化状態への遷移は、暗号鍵有効期間が満了したなどの予め決められた条件に従って実行される。一方、危殆化状態への遷移は、暗号鍵の漏えいなどによる予期せぬ原因により、当該暗号鍵の安全性が担保できないと判断された場合に実行される。つまり、暗号鍵の安全性低下が遷移の主要因になっているかどうかが､非活性化状態への遷移と危殆化状態への遷移の違いである。
鍵失効機能は、危殆化状態への遷移を実行するための機能のことである。

項目 B.13 は、どのようなことが発生したら危殆化状態と見なすのかといった鍵失効の遷移条件と誰にどのように危殆化状態に遷移した事実を知らせるのかといった通知方法など、鍵失効機能への要求事項を明確化することを求めている。これは、B.03 の危殆化状態への遷移条件と整合していなければならない。B.14 は、実際に利用する具体的な鍵失効メカニズム及び通知メカニズムを明確化することを要求している。

鍵失効の遷移条件及び具体的な鍵失効メカニズムは、「いつ、どのように、どのような状況で危殆化した（と推定される）鍵情報を失効させ、利用できなくするか」の視点で検討される。

一方で、この機能が実行されると、それ以降の保護（暗号化や署名生成）のセキュリティが担保されないだけでなく、過去に保護された情報の処理（復号や署名検証）に対しても完全なセキュリティは保証されなくなる。このため、危殆化した暗号鍵を速やかに置き換え、新しい暗号鍵で保護が再開できるようにすることはもとより、当該暗号鍵を使用して過去に保護された情報を処理するエンティティ（利用者）全員に完全なセキュリティが保証されない可能性があることの注意喚起を行う必要がある。鍵失効の通知が重要であるのはこのためである。

失効情報を依拠する当事者及び依拠するエンティティへの通知メカニズムは、「危殆化した（と推定され、失効した）暗号鍵を使用して過去に保護された情報について完全なセキュリティが保証されない可能性があることをどのように関係者に通知するか」の視点で検討される。例えば、危殆化／失効の通知方法としては、危殆化鍵リスト、証明書失効リスト（CRLs)、ホワイトリスト、クエリホワイトリスト、OCSP（Online Certificate Status Protocol）がある。

⑥　暗号鍵の一時停止機能及び再活性化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.15	FR6.29	CKMS 設計は、どのように、どのような状況で鍵が一時停止されるかを明記しなければならない。	6.4.6 節
B.16	FR6.30	CKMS 設計は、どのように一時停止情報を依拠又は通信する当事者が利用可能になるかを明記しなければならない。	6.4.6 節
B.17	FR6.32	CKMS 設計は、どのように一時停止された鍵によるセキュリティ処理（services）の実行を防止するのかを明記しなければならない。	6.4.6 節
B.18	FR6.31	CKMS 設計は、どのように、どのような状況で一時停止された鍵が再活性化されるかを明記しなければならない。	6.4.6 節
B.19	FR6.33	CKMS 設計は、どのように再活性化情報を依拠又は通信する当事者が利用可能になるのかを明記しなければならない。	6.4.6 節

解説・考慮点

暗号鍵の一時停止状態への遷移を提供する一時停止機能、及び活性化状態への遷移を再度提供する再活性化機能のことであり、これらの機能は必ずセットで用いられる。
項目 B.15 及び B.16 は、CKMS の設計にあたって、暗号鍵の一時停止の遷移条件及び通知方法といった一時停止機能への要求事項を明確化することを求めたものである。B.17 は、一時停止された暗号鍵が利用されないようにするための要求事項を明確化することを求めたものである。B.18 及び B.19 は、暗号鍵を再活性化するための遷移条件及び通知方法といった再活性化機能への要求事項を明確化することを求めたものである。

なお、鍵状態において一時停止状態を定義しない場合には、これらの項目は対象外である。

一時停止状態は、暗号鍵の利用が一時的に認められなくなっている状態であり、i）暗号鍵の危殆化が疑われ、その状況を調査するための時間を確保しつつ、万が一の危殆化の影響が発生しないようにするケースと、ii）当該暗号鍵を所有するエンティティが一定期間利用しない（長期休暇など）ことが明らかであり予防保全を行うケース、のどちらかの理由で使われることが想定されている。前者の「暗号鍵の危殆化の疑い」には、暗号鍵の誤使用や誤配置などに暗号鍵の危殆化につながりかねない状況を含んでもよい。

なお、鍵状態として必ず一時停止状態を用意する必要があるわけではなく、i）のケースでは活性化状態から危殆化状態に直接遷移するようにし、ii）のケースでは活性化状態のままにしておく、という管理の仕方をしても構わない。もし一時停止状態を定義しない場合には、ここでの項目は全て対象外となる。

一時停止状態を用意することのメリットは、i）のケースでは、調査の結果で危殆化の恐れが払拭された場合には、同じ暗号鍵を再活性化することで継続利用が可能になり、暗号鍵の更新に関連する処理を実施する必要はないことである。もし危殆化状態に遷移させていた場合、危殆化の恐れが払拭された場合でも暗号鍵の更新・再設定が必要となる。また、ii）のケースでは、長期利用しないエンティティになりすまされて不正利用される事態を防止できることである。

一方、一時停止状態と用意する場合には、どのようなことが発生したら一時停止させるのかといった一時停止状態への遷移条件（図 2-5 の⑦）と、どのようなことが満たされたら一時停止を解除するのかといった活性化状態への遷移条件（図 2-5 の⑩）を必ずセットで準備しなければならない。また、一時停止状態になった暗号鍵は保護（暗号化や署名生成）を行うために使用できないので、当該暗号鍵を利用する処理が行われることはないことを関係するエンティティ（利用者）全員に注意喚起する必要がある。

項目 B.15 と B.18 は上述の一時停止状態への遷移条件と活性化状態への遷移条件を、B.16 と B.19 は関係エンティティ(利用者)への通知方法を明確化することを求めたものである。B.17 は、一時停止された暗号鍵が利用されないようにするための要求事項の明確化のことである。これらの要件は B.03 の一時停止状態への遷移条件及び再活性化の遷移条件と整合していなければならない。

⑦　鍵情報の破壊機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.20	FR6.38	CKMS 設計は、どのように、どのような条件で鍵が意図して破壊されるか、及び破壊がコンポーネントへの局所的（local）なものであるか CKMS 全体への共通的（universal）なものであるかを明記しなければならない。	6.4.9 節

B.21	FR6.39	それぞれの鍵タイプに対して、CKMS 設計は、鍵破壊の事前通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の時期が含まれる。	6.4.9 節

解説・考慮点

暗号鍵の破壊状態への遷移を提供する機能である。
項目 B.20 は、CKMS の設計にあたって、鍵情報を破壊するための条件及び具体的な破壊方法、並びに当該鍵情報がどこに存在するかを含めて明確化することを要求したものである。B.21 は、鍵情報が破壊されたことの通知方法といった破壊機能への要求事項を明確化することを求めたものである。

暗号鍵が使えない状態になっていたとしても、どこかに当該暗号鍵のデータが残っていれば、攻撃や不正持出など、何らかの理由によりそのデータが漏えいするリスクを完全には排除できない。逆に言えば、暗号鍵の漏えいリスクを完全に排除するためには、当該暗号鍵が完全に存在しなくなった状況にするしかない。当然、実際に利用中の暗号鍵だけでなく、バックアップやアーカイブされたものも含めて完全に存在しない状況にする必要がある。つまり、バックアップやアーカイブしている場合で鍵の破壊を行うケースでは、鍵の破壊[8]に関して B.20 の記載内容と B.54 や B.56 の記載内容が整合していなければならない。

一方、暗号鍵を破壊してしまえば、例え当該暗号鍵の所有者であったとしても、当該暗号鍵で保護（暗号化）を行った情報を処理（復号）することはできなくなり、その情報は永遠に失われる結果を招くことになる。したがって、誤って暗号鍵の破壊をしてしまうことがないようにしておくことも重要である。
項目 B.20 では、CKMS の設計にあたって、鍵情報を破壊するための条件などの他、破壊した暗号鍵の影響範囲がどこまで及ぶのかを明確にしておくことを求めている。例えば、利用者 A だけが使う暗号鍵であれば、その暗号鍵が破壊されたことによって影響を受けるのは利用者 A だけに限定される。つまり、局所的（local）な影響である。一方、CKMS 全体を管理するルート鍵が破壊されると CKMS 全体で当該ルート鍵が利用できなくなるので、その影響は共通的（universal）であると言える。
なお、ここでの要件は B.03 の破壊状態への遷移条件と整合していなければならない。

B.21 は、鍵情報が破壊されたことの通知方法の要求事項を明確化することを求めたものである。鍵情報の破壊は、当該暗号鍵が使えなくなるだけでなく、当該暗号鍵で保護（暗号化）を行った情報も喪失することを意味するので、注意喚起の意味でも関連するエンティティ（利用者）に事前通知しておくことが特に重要になる。

[8] SP 800-88 Rev.1「Guidelines for Media Sanitization（媒体のデータ抹消処理（サニタイズ）に関するガイドライン）」中に暗号鍵の破壊方法についての記載（5 章参照）があるので、必要に応じて参照されたい。
https://csrc.nist.gov/publications/detail/sp/800-88/rev-1/final
日本語訳：https://www.ipa.go.jp/security/crypto/archive.html

⑧　鍵生成機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.22	FR6.19	CKMS 設計は、それぞれの鍵タイプに対して、CKMS で使用される鍵生成手段を明記しなければならない。	6.4.1 節
B.23	FR6.20	CKMS 設計は、対称鍵及びプライベート鍵を生成するのに使用される元となる乱数生成器を明記しなければならない。	6.4.1 節

解説・考慮点

CKMS の設計にあたって、項目 B.22 は暗号鍵を生成する手段について、B.23 は暗号鍵を生成する際に利用する乱数生成器について明確化することを要求したものである。一般に、鍵生成手段は暗号鍵と対になる暗号アルゴリズムの仕様に依存し、暗号目的として設計された乱数生成器の使用を要求する。

暗号アルゴリズムの安全性は、利用する暗号鍵が予測できないことを前提としている。逆に言えば、利用する暗号鍵が何らかの方法で予測できるのであれば、どんなに強力な暗号アルゴリズムを使っていたとしても安全性は担保されない。そのため、予測できない暗号鍵を生成する手段を利用することが極めて重要になる。

ここでの「予測できない」の意味は、①出力が一様分布になっていること、②十分なエントロピー量が確保されていること、の両方を満たすことである。

①の出力が一様分布になっているということは、生成される可能性がある暗号鍵の種類が 2^{128} 個あるならば、どの鍵も一様に $1/2^{128}$ の確率で生成されるということである。このような性質を持つように作られていると信頼できる鍵生成方法として、SP 800-90 や SP 800-133 などの乱数生成器を使う方法と、SP 800-56 などのような鍵導出関数を使う方法がある。

②の十分なエントロピー量を確保されているということは、極めて多数の種類の暗号鍵が生成可能であるということを意味する。例えば、鍵長 128 ビットの暗号鍵であれば 2^{128} 個の異なる暗号鍵が利用できることが期待されるので、利用する鍵生成手段が 2^{128} 個の異なる暗号鍵を生成するように作られていれば「十分なエントロピー量を確保されている」という。一方、例えば、1,000 個の異なる暗号鍵しか生成しないような鍵生成手段であれば、例え出力が一様分布になっていたとしても「エントロピー量が十分に確保されていない」と判断される。これは、暗号鍵の全数探索で暗号解読しようとした場合、前者は平均 2^{127} 回のトライアルが必要となるが、後者は平均 500 回のトライアルをすればよいためである。

項目 B.22 は暗号鍵を生成する手段について、B.23 は暗号鍵を生成する際に利用する乱数生成器について明確化することを要求したものである。ここでのポイントは「信頼できる鍵生成手段」や「信頼できる乱数生成器」を使い、十分なエントロピー量が確保されていることを確認することにある。

⑨　鍵導出機能／鍵更新機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.24	FR6.36	CKMS 設計は、鍵を導出又は更新するために使用される全てのプロセス、及び鍵が導出又は更新される状況を明記しなければならない。	6.4.8 節
B.25	FR6.37	それぞれの鍵タイプに対して、CKMS 設計は、鍵の導出又は更新の事前通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.8 節

解説・考慮点

暗号鍵の生成では、鍵生成機能を利用する代わりに、鍵導出機能や鍵更新機能を利用することがある。
項目 B.24 及び B.25 は、CKMS の設計にあたって、鍵導出機能／鍵更新機能が利用される条件や鍵導出方法、通知方法等といった鍵導出機能／鍵更新機能への要求事項を明確化することを求めたものである。
なお、これらの機能は採用必須の機能ではなく、CKMS の設計で採用しなければ検討対象外である。逆に、採用してはならないという要求事項の場合もあり得る。

鍵導出機能では、一部が秘密であるような独立した他の情報（他の暗号鍵、共有秘密やパスワードなど）から不可逆な形で暗号鍵が導出されるプロセスを実行する。例えば、鍵確立プロトコルでは互いの共有秘密から共有鍵を導出する。

鍵更新機能では、「元鍵」から「別鍵」を計算で導出し、導出した「別鍵」で「元鍵」を置き換えるプロセスを実行する。なお、別鍵を導出する際に他の秘密データを使用しない場合には、元鍵と更新方法を知っている攻撃者が将来にわたるあらゆる時期の更新した別鍵を知りうるというセキュリティリスクにさらされる。

暗号鍵の予測可能性の観点でいえば、二つの点で、鍵生成機能を利用して生成される暗号鍵よりも予測がしやすくなっている可能性がある。①鍵導出機能や鍵更新機能により作られる暗号鍵は、「（乱数生成器ではなく）ある種の計算方法」に従って生成されることから、入力データと出力データ（生成された暗号鍵）との間に相関が残っている可能性がある、②入力データの種類によっては十分なエントロピー量が確保できない可能性がある。例えば、パスワードなどから鍵を導出するからといって、十分なエントロピー量の確保できるような複雑なパスワードを利用させることは困難を伴う。

これらの弱点による暗号鍵の予測可能性を少しでも低減するには、信頼できる鍵導出方法や鍵更新方法を使うのが望ましい。例えば、SP 800-108 や SP 800-132、SP 800-135 などである。

一方、鍵導出機能や鍵更新機能を使うことのメリットは、i）ローカルで（鍵生成機能を使うことなく）鍵更新ができる、ii）鍵確立プロトコルを利用しなくてもそれぞれのエンティティ（利用

者）が独自に同じ秘密データを使って同じ暗号鍵を生成することができる、という点である。前者は特に大規模 CKMS の場合や CKMS につながっていない機器などの場合に効果を発揮する可能性があり、後者は鍵確立時の漏えいリスクの低減に役立つ。

項目 B.24 及び B.25 は、CKMS の設計にあたって、鍵導出機能／鍵更新機能が利用される条件や鍵導出方法、通知方法等といった鍵導出機能／鍵更新機能への要求事項を明確化することを求めたものである。ここでのポイントは、B.24 は主に「信頼できる鍵導出方法や鍵更新方法」を使うことで暗号鍵の予測可能性のリスクを低減していることの確認を行うことである。B.25 は鍵導出や鍵更新を行う事前通知がないまま新しい暗号鍵に変わった場合、当該暗号鍵を利用するエンティティ（利用者）にとっては突然以前の暗号鍵が使えなくなることを意味し、それが故障などの予期せぬ原因によって生じた事態だと誤認する恐れがある。このような事態を避けるのが事前通知の役割であり、何らかの方法で通知する仕組みを設けたほうがよい。ただし、通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

なお、鍵導出機能や鍵更新機能は採用が必須の機能ではないので、CKMS 設計でこれらの機能を採用しないと決めたのであれば検討対象外となる。さらには、採用してはならないという要件を採用する場合もあり得る。

⑩　対称鍵の検証機能への要求事項／⑪　公開鍵の検証機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.26	FR6.66	CKMS 設計は、どのように、どこで、どのような状況で、対称鍵やそのメタデータが検証されるかを明記しなければならない。	6.4.24 節
B.27	FR6.63	CKMS 設計は、どのように、どこで、どのような状況で、公開鍵ドメインパラメタが検証されるかを明記しなければならない。	6.4.21 節
B.28	FR6.64	CKMS 設計は、どのように、どこで、どのような状況で、公開鍵が検証されるかを明記しなければならない。	6.4.22 節
B.29	FR6.65	CKMS 設計は、どのように、どこで、どのような状況で公開鍵証明書パスが検証されるかを明記しなければならない。	6.4.23 節

解説・考慮点

＜対称鍵の検証機能＞
対称鍵及びそのメタデータに対するテストを実行する機能である。
項目 B.26 は、CKMS の設計にあたって、対称鍵の検証機能が利用される条件や検証方法等といった対称鍵の検証機能への要求事項を明確化することを求めたものである。

＜公開鍵の検証機能＞
公開鍵についてある種の正当性チェックを実行して公開鍵が数学的に正しいことを保証し、公開ドメインパラメタについてもドメインパラメタが数学的に正しいことの保証を提供する機能である。
CKMS の設計にあたって、項目 B.27～B.29 は、公開鍵の検証機能が利用される条件や検証方法等といった公開鍵の検証機能への要求事項を明確化することを求めたものである。

生成されたり、共有されたりした暗号鍵が正当なものであることは、暗号アルゴリズムを利用する上での前提条件である。したがって、実際の暗号処理に利用する前には正当性を確認しておくことが必要である。

ただし、その確認を行うタイミングは、利用する暗号鍵の種類や利用方法、利用環境などによってさまざまである。例えば、タイミングだけ見ても、①暗号鍵が生成されたタイミングで実施、②暗号処理を行う直前に実施、③暗号鍵が活性化状態に遷移するタイミングで実施、④定期的に実施、などが考えられる。検証方法についても、a）ハッシュ値でのチェック、b）署名でのチェック、c）物理的手段でのチェック、d）相互通信可否でのチェック、e）CKMS サーバでのチェック、などが考えられる。

項目 B.26～B.29 は、いずれも検証機能が利用される条件や検証方法等といった、暗号鍵の検証に必要な要求事項を明確化することを求めている。具体的には、B.26 では対称鍵とそれに関連するメタデータが、B.27 では B.28 での公開鍵が利用するドメインパラメタが、B.28 では公開鍵が、それぞれ検証の対象として、「いつ（タイミング）、どこで（検証場所）、どのように（検証方法）」検証を行うのかを洗い出すことである。

同様に、B.29 では、B.28 での公開鍵とセットで作られる公開鍵証明書について、トラストアンカーから始まる公開鍵証明書のチェーン(証明書パス)を使った有効性検証を対象としている。

⑫　トラストアンカー管理機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.30	FR6.70	CKMS 設計は、サポートされている全てのトラストアンカー管理機能を明記しなければならない（[RFC6024] を参照)。	6.4.28 節
B.31	FR6.71	CKMS 設計は、依拠するエンティティがトラストアンカーについてのソース認証（source authentication）及び完全性検証を実行できるように、どのようにそれらのトラストアンカーがセキュアに配付されるかを明記しなければならない。	6.4.28 節
B.32	FR6.72	CKMS 設計は、依拠するエンティティのシステムのトラストアンカーストアに対して、認可された追加、変更、削除のみが行えること	6.4.28 節

		を保証するために、どのように依拠するエンティティのシステムでトラストアンカーが管理されるかを明記しなければならない。	

解説・考慮点

トラストアンカーなしでは信頼できるかわからない公開鍵に対して、信頼を確立するために使用されるトラストアンカーを保管・管理する機能である。
CKMS の設計にあたって、項目 B.30 はトラストアンカー管理機能についてどのようなものを受け入れるのかといった要求事項を、B.31 及び B.32 はトラストアンカーを完全かつセキュアに配送・保管・追加・削除等といったメンテナンスを行うためのトラストアンカー管理機能への要求事項を明確化することを求めたものである。

公開鍵暗号・署名では、プライベート鍵とそれに対応する公開鍵は一対一対応しているが、問題はそのプライベート鍵の所有者が「正当な」エンティティであることの保証が「公開鍵」からは得られないことである。例えば、Eve が自分自身のプライベート鍵に対応する公開鍵を「Alice の公開鍵」であると偽って公開していた場合に、Bob がその公開鍵が「Alice のものではない」と見破ることはほとんどできない。つまり、何も対策をしていないままの公開鍵は基本的に「信頼できるかわからない公開鍵」である。

そこで、信頼できる第三者であるトラストアンカーを用意し、そのトラストアンカーが「信頼できるかわからない公開鍵」にお墨付きを与えることで「信頼できる公開鍵」にする仕組みが必要になり、それを実現したのが公開暗号基盤（PKI）である。つまり、PKI では、トラストアンカーが公開鍵の信頼性の起点となることを意味し、その信頼性のうえで公開鍵暗号・署名の安全性が確保されていることになる。したがって、トラストアンカーの完全性確保は CKMS のセキュリティにとって死活的に重要である。

項目 B.30～B.32 は PKI でのトラストアンカーの完全性確保のための要求事項を明確化することを求めたものである。ここでのポイントは、具体的に、B.30 はどんなトラストアンカーを PKI の利用エンティティのシステムに用意するのか、B.31 はそのシステムに対してトラストアンカーの情報をどのように安全に配布するのか、さらに B.32 はそのシステムでのトラストアンカーの情報をどのように安全に管理するのか、といったことを確認することである。

例えば、トラストアンカー管理機能として OS での証明書管理機能を使い（B.30）、CKMS サーバから準備するトラストアンカーの公開鍵証明書（プライベート CA ルート証明書）を当該 CKMS サーバからダウンロードさせ、パスワード及びハッシュ値チェックで OK になった場合にそのルート証明書を証明書管理機能が管理するトラストアンカーストアに登録し（B.31）、OS での証明書管理機能でルート証明書の管理を行う（B.32）といったことである。

⑬　公開鍵の有効期間延長機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.33	FR6.34	CKMS 設計は、どのように、どのような条件で公開鍵の有効期間が延長できるかを明記しなければならない。なお、延長後の有効期間の合計が CKMS セキュリティポリシー等で定める公開鍵の最大許容暗号鍵有効期間を超える場合には、当該公開鍵の延長を行ってはならない。	6.4.7 節
B.34	FR6.35	それぞれの鍵タイプに対して、CKMS 設計は、鍵タイプの有効期間延長の事前通知の要求事項を明記しなければならない。それには、どの当事者に通知されるか、どのように通知されるか、どのセキュリティ処理（services）が通知に適用されるか、及び通知の期間が含まれる。	6.4.7 節

解説・考慮点

新しい有効期限を設定した「同じ公開鍵」を含む新しい公開鍵証明書を発行することで、以前の有効期間を超えて既存の公開鍵に対する新しい有効期間を確立するための機能である。なお、延長後の有効期間の合計が CKMS セキュリティポリシー等で定める公開鍵の最大許容暗号鍵有効期間を超える場合には、当該公開鍵の延長を行ってはならない。
CKMS の設計にあたって、項目 B.33 及び B.34 は、公開鍵の有効期間延長機能が利用される条件や事前通知方法等といった公開鍵の有効期間延長機能への要求事項を明確化することを求めたものである。

セキュリティ上の観点から、公開鍵証明書には有効期間が記載されている。有効期間が経過すると非活性化状態に自動的に遷移して当該公開鍵は失効し、また新しい公開鍵が鍵生成機能を使って生成されるのが一般的である。この場合、B.33 と B.34 は検討対象外となる。

しかし、新しい公開鍵を生成した場合、関連する鍵情報の更新も合わせて必要となったり、過去に署名したものに対する署名検証ができなくなったりするといったデメリットもある。こういったデメリットを回避する方法として、当該公開鍵の有効期間を延長するというやり方がある。これは、公開鍵の有効期間延長機能を使って、新しい有効期間を設定した「同じ公開鍵」を含む新しい公開鍵証明書を発行することによる実現する。

そもそも公開鍵証明書の有効期間は、トラストアンカーが「信頼できるかわからない公開鍵」を「信頼できる公開鍵」として利用できるお墨付きを与えている期間である。言い換えれば、その期間内は、公開鍵の安全性の担保をトラストアンカーが代理で受け持つということである。このことを踏まえれば、「有効期間を延長」することができるかどうかはトラストアンカーがその分の「責任を継続して負う」ことができるかどうかに依存する。項目 B.33 は、この「責任を継続して負う」ことができる条件を明確化することを求めたものである。

なお、CKMS セキュリティポリシー等で定める公開鍵の最大許容暗号鍵有効期間は、セキュリ

ティ上の要件として定められたものであるので、これに違反するような延長は認められないことに注意されたい。例えば、最大許容暗号鍵有効期間が 3 年である場合、1 年間有効の公開鍵証明書は 2 度延長することはできるが、3 度目の延長はできない。

公開鍵証明書の有効期間を延長するためには、その有効期間が切れる前に新たな有効期間を設定した公開鍵証明書の発行が必要となる。その手続きを行わせるためのトリガーとなるのが事前通知であり、項目 B.34 で事前通知を行うための条件の明確化を求めている。通知手段としては、利用者にも知らせる形で行われる場合もあれば、利用者に感知させることなく機器間で自動的に行われる場合もある。

⑭　所有者登録機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.35	FR6.21	CKMS 設計は、鍵と所有者の識別子を結び付けるプロセスを含めて、所有者登録に関わる全てのプロセスを明記しなければならない。	6.4.2 節

解説・考慮点

セキュリティエンティティ（個人、組織、デバイス、又はプロセス）及びメタデータを伴う暗号鍵の最初の登録を行うための機能である。
項目 B.35 は、CKMS の設計にあたって、所有者登録機能への要求事項を明確化することを求めたものである。

暗号鍵は、正しいエンティティ（利用者）に届ける必要がある。もし登録時点で誤った所有者登録が行われてしまうと、誤ったエンティティと暗号鍵が結び付くことになり、その後の暗号鍵のライフサイクルが正しく実行されたとしても全く安全性が担保されない。

したがって、所有者登録では、①どのようにエンティティ（利用者）が正しいことを確認するか、②どのようにそのエンティティと暗号鍵とを結びつけるか、を把握しておくことが重要である。典型的には、エンティティの対称鍵、公開鍵又はプライベート鍵の初期セットと、エンティティ識別子及びメタデータとも結び付ける登録プロセスが存在する。

項目 B.35 は、これら把握しておくべきことを明らかにすることを求めている。

⑮　プライベート鍵所持の検証機能への要求事項／⑯　プライベート鍵の検証機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.36	FR6.68	CKMS 設計は、どのように、どこで、どのような状況で、プライベート鍵とそのメタデータの所持が検証されるかを明記しなければならない。	6.4.26 節
B.37	FR6.67	CKMS 設計は、どのように、どこで、どのような状況で、プライベート鍵又は鍵ペア、あるいはそのメタデータが検証されるかを明記しなければならない。	6.4.25 節

解説・考慮点

＜プライベート鍵所持の検証機能＞
公開鍵の所有者であると主張する者が対応するプライベート鍵を所持していることの保証を得るために、公開鍵を受領したエンティティによって使用される機能である。
項目 B.36 は、CKMS の設計にあたって、プライベート鍵所持の検証機能が利用される条件や検証方法等といったプライベート鍵所持の検証機能への要求事項を明確化することを求めたものである。

＜プライベート鍵の検証機能＞
プライベート鍵に対してある種のテストを実行し、鍵ペアの仕様を満たすことの保証を提供するための機能である。
項目 B.37 は、CKMS の設計にあたって、プライベート鍵の検証機能が利用される条件や検証方法等といったプライベート鍵の検証機能への要求事項を明確化することを求めたものである。

PKI に則り、トラストアンカーが「信頼できるかわからない公開鍵」を「信頼できる公開鍵」として利用できるお墨付きを与えるためには、その公開鍵に対応する「正しい」プライベート鍵（及びそれに付随するメタデータ）を「正当な」エンティティが所有していることを確認する必要がある。

その一方、プライベート鍵は基本的に唯一のエンティティのみが秘密に所持することが求められるため、当該公開鍵に対応する「正しい」プライベート鍵（及びそれに付随するメタデータ）を「正当な」エンティティが所有していることを、トラストアンカーにその中身を直接示して証明するわけにはいかない。

そこで、項目 B.36 は、プライベート鍵（及びそれに付随するメタデータ）（と称するデータ）を所持していることを「正しい」エンティティが所有していることを確認するための検証方法の明確化を求めたものである。また、B.37 は B.36 での検証対象となったプライベート鍵（及びそれに付随するメタデータ）（と称するデータ）と（お墨付きを与える対象の）公開鍵とが一対一の

正しい関係にあり、結果として「正しい」プライベート鍵（及びそれに付随するメタデータ）であることを確認するための検証方法の明確化を求めたものである。

つまり、B.36 及び B.37 に両方を実施することで、トラストアンカーにプライベート鍵（及びそれに付随するメタデータ）の中身を直接示すことなく、当該公開鍵に対応する「正しい」プライベート鍵（及びそれに付随するメタデータ）を「正当な」エンティティが所有していることを証明する。なお、これらの機能は、プライベート鍵の所有者又はプライベート鍵の所有者の代理として振舞う信頼される第三者のみが実行できる。

⑰　暗号鍵とメタデータの関連付け機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.38	FR6.40	使用されているそれぞれの鍵タイプに対して、CKMS 設計は、何のメタデータが鍵と関連付けられているか、どのようにメタデータが鍵と関連付けられているか、及びメタデータが鍵と関連付けられる状況を明記しなければならない。	6.4.10 節
B.39	FR6.41	使用されているそれぞれの鍵タイプに対して、CKMS 設計は、どのように次のセキュリティ処理（services）（保護）が関連付けられたメタデータに適用されるかを明記しなければならない：ソース認証（source authentication）、完全性、及び機密性。	6.4.10 節

解説・考慮点

暗号鍵に関連付けられているメタデータ要素がある場合、それらを関連付けるための機能である。なお、暗号鍵とメタデータの関連付けは、当該鍵情報の生成時のほか、配送時、登録時、保管時など、暗号鍵有効期間を通じて完全性を維持する必要がある。
CKMS の設計にあたって、項目 B.38 及び B.39 は、関連付け機能が利用される対象や条件、関連付けの方法等といった関連付け機能への要求事項を明確化することを求めたものである。

鍵情報は、暗号鍵とメタデータとで構成される。4.1 節に記載があるように、メタデータとは、暗号鍵を適切に管理するために、その暗号鍵に関連付けられている情報である。例えば、暗号鍵のパラメタ、保護方法や有効期間などである。「暗号鍵管理システム設計指針（基本編）」では、メタデータの典型的な要素として 23 種類が記載されている。鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。

項目 B.38 と B.39 は、各々の鍵タイプに対して、暗号鍵に関連付けられているメタデータがあるのか、ある場合にはどのようなメタデータが関連付けられているのか、どのように関連付けられているのか、そして、何のためにそれが行われているのか、を把握することにポイントがあり、それを実現するために必要な機能要件を明確化することである。

ちなみに、関連付けを提供する保護メカニズムには、暗号学的プロセスを使用する場合と信頼プロセスを使用する場合とがある。直感的には、前者は、暗号鍵とメタデータの組で計算されたデジタル署名など、暗号アルゴリズムによって関連付けが保証される。後者は、信頼されるエンティティからのメタデータの対面手渡しやセキュアなストレージでの保管など、物理的な手段で関連付けが保証される。

留意するポイントとしては、B.38 と B.39 は「関連付け機能への要求事項」であるため、詳細な内容までを求めているわけではなく、そういった具体的な内容は D.02 以降で取りまとめられる。B.38 と B.39 は、D.02 以降に記載された暗号鍵とメタデータの関連付けを行う上で必要となる機能が用意されているか、整合的であるかの観点で確認することが重要である。

⑱　メタデータの変更機能への要求事項／⑲　メタデータの削除機能への要求事項／⑳　暗号鍵のメタデータリスト化機能への要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.40	FR6.42	CKMS 設計は、関連付けられたメタデータが変更される状況を明記しなければならない。	6.4.11 節
B.41	FR6.43	CKMS 設計は、鍵と関連付けられたメタデータが削除される状況を明記しなければならない。	6.4.12 節
B.42	FR6.44	CKMS 設計は、関連付けられたメタデータを削除するために使われる手法を明記しなければならない。	6.4.12 節
B.43	FR6.45	それぞれの鍵タイプに対して、CKMS 設計は、どのメタデータが認可されたエンティティによってリスト化が可能かどうかを明記しなければならない。	6.4.13 節

解説・考慮点

＜メタデータの変更機能＞
認可されたエンティティが、暗号鍵と関連付けられている既存の書き込み可能なメタデータを変更するために使用する機能である。
項目 B.40 は、CKMS の設計にあたって、メタデータの変更機能が利用できる対象や条件、認可されていないエンティティの利用防止策等といったメタデータの変更機能への要求事項を明確化することを求めたものである。

＜メタデータの削除機能＞
認可されたエンティティが、暗号鍵に関連付けられたメタデータを削除するために使用する機能である。
CKMS の設計にあたって、項目 B.41 は、メタデータの削除機能が利用できる対象や条件、認

可されていないエンティティの利用防止策等といったメタデータの削除機能への要求事項を明確化することを求めたものである。B.42 は、具体的な削除方法の明確化することを要求したものである。

＜暗号鍵のメタデータリスト化機能＞
エンティティに認可されている暗号鍵のメタデータのリスト化を当該エンティティが実行するための機能である。
項目 B.43 は、CKMS の設計にあたって、メタデータリスト化機能が利用できる対象や条件といったメタデータリスト化機能への要求事項を明確化することを求めたものである。

前の要求事項でも記載したように、鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。つまり、暗号鍵だけでなく、メタデータも不用意に変更されたり削除されたりしないことが必要である。
そのためには、メタデータの完全性に影響するような処理を行えるエンティティを制限し、かつ認可されたエンティティであっても許可された範囲内、例えば自分の管理下にあるメタデータに対してのみ変更や削除、参照が行えるようにしておくことが望ましい。

項目 B.40 は変更権限を持つエンティティのみが決められた範囲内で変更機能を利用できるように、B.41 と B.42 は削除権限を持つエンティティのみが決められた範囲内で決められた方法による削除機能を利用できるように、それぞれのアクセス制御を行うために必要な要求事項の明確化を求めている。B.43 は、メタデータを一覧として参照できる権限を持つエンティティが決められた範囲内のメタデータのみ参照できるようにアクセス制御を行うために必要な要求事項の明確化を求めている。

《トイモデルと記載例》

本節のトイモデルも、2.1 節のトイモデルと同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節のトイモデルで定めたものとする。
また、システムの運用条件を以下のように設定する。【】内は記載例のどの項目に影響を与えているのかを示している。

- 鍵生成はメール送信 PC において信頼できる方法で生成し、プライベート鍵はライフサイクル全般を通じて PC 外部に複製されることはない。【B.04, B.22, B.23】
- CA から公開鍵証明書を受信したときに、当該証明書の正当性を確認する。【B.27, B.28】
- 公開鍵証明書の有効期間が始まった鍵は自動的に活性化状態となる。【B.04, B.06, B.07】
- 署名の生成や検証は、メール送受信 PC のメール管理部でのみ行う。【B.08】
- 署名付きメールを受信したとき、受信した公開鍵証明書の正当性を検証する。【B.04, B.27～B.29, B.36】

- 公開鍵証明書の有効期限切れした鍵は、自動的に非活性化状態になる。【B.04, B.09～B.11】
- 鍵の所有者が鍵を必要ないと判断したとき、手動で削除する。【B.04, B.20, B.21】
- 情報システム部を介してパブリックな CA 局に署名を依頼することで、信頼できる証明書チェーンを構築し、基本的なトラストアンカー管理は更新機能など OS の機能より実現する。【B.04, B.05, B.29～B.32, B.37】
- 公開鍵証明書の運用管理は、情報システム部の担当者が行う。【B.12, B.33～B.35】
- 鍵の危殆化が疑われるときは失効処理を行う。【B.04, B.13, B.14】
- 鍵のバックアップ、アーカイブは行わない。【B.04】
- 鍵とメタデータの関連付けは公開鍵証明書により行う（暗号学的プロセス）。【B.38, B.39】
- メタデータの変更・削除・リスト化は認めない。【B.11, B.33, B.40～B.43】
- 鍵の一時停止状態は設定しない。【B.04, B.15～B.19】
- 鍵導出機能や鍵更新機能は使用しない。【B.04, B.24, B.25】
- 対象鍵は使用しない。【B.04, B.26】

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.04	● メール送信者 PC の鍵生成機能により、署名プライベート鍵と署名公開鍵を生成する。 ● 公開鍵証明書の有効期間が開始したら、鍵活性化機能により、署名プライベート鍵と署名公開鍵を活性化状態にする。 ● メール送信者 PC の暗号機能により、メールのドキュメントに署名する。 ● メールを受信したら、メール送信者から送られてきたことを確認するために、メール受信者 PC の暗号機能により署名の完全性を確認する。 ● 公開鍵の検証機能により、署名公開鍵に対する公開鍵証明書に対する完全性を確認し、公開鍵及びパラメタの検証を行う。 ● OS でのトラストアンカー管理機能により、ルート CA の公開鍵証明書を保管・管理する。情報の更新は、OS 又はブラウザの自動アップデートにより実施する。 ● 公開鍵証明書の有効期間が終了したら、鍵非活性化機能により、署名プライベート鍵と署名公開鍵を非活性化状態にする。 ● 署名プライベート鍵と署名公開鍵は、破壊条件を満たした場合、破壊機能により、鍵を破壊する。 ● 署名プライベート鍵の危殆化が疑われるときは、鍵失効機能により、署名プライベート鍵の失効処理を行う。署名公開鍵についても同様の処理を行う。 ● 利用者の管理は情報システム部が行うものとし、そのために必要な管理機能は情報システム部管理の機器により実現する。 ● 暗号鍵とメタデータの検証及び関連付けについては、公開鍵証明書の申請段階で情報システム部がその正当性を検証するものとし、そのために必要な管理機能は情報システム部管理の機器により実現する。

	参考： ● 一時停止状態は設定しないので、一時停止機能及び再活性化機能は使わない。 ● 鍵生成では、鍵生成機能のみを利用するものとし、鍵導出機能／鍵更新機能は使わない。 ● 対称鍵の生成・利用は行わないので、対称鍵の検証機能は使わない。 ● 公開鍵の有効期間延長は認めないので、公開鍵の有効期間延長機能は使わない。 ● 署名プライベート鍵が利用できなくなることによる影響は当該鍵の所有者だけであり、影響が局所的であるので、バックアップとアーカイブの処理は実施せず、これらの機能は使わない。
B.05	● 署名公開鍵と関連メタデータの完全性は、公開鍵証明書の CA 署名検証により確認する。 ● 署名プライベート鍵の機密性は、OS のファイルアクセス機能により当該鍵を作成したユーザ以外が鍵ファイルにアクセスできないように管理することで実現する。 ● CA の署名公開鍵は OS の信頼できる公開鍵証明書（トラストアンカー）からなるチェーンの有効性により完全性を確認する。 ● OS の信頼できる公開鍵証明書（トラストアンカー）の更新は、OS 又はブラウザの自動アップデートにより実行される。
B.06	CA の署名した公開鍵証明書が、OS の信頼できる公開鍵証明書（トラストアンカー）からなるチェーンの有効性確認で正当であり、かつ記載された有効期間になったら、署名プライベート鍵と署名公開鍵は活性化状態に遷移する。
B.07	鍵活性化の通知は行わない。 メール送信者は CA が署名した公開鍵証明書の有効期間により活性化状態であることを認識できる。メール受信者は送信者の公開鍵証明書が有効期間内であることにより活性化状態であることを認識できる。
B.08	メールの署名は、メール送信 PC のメール管理部の管理下で実行される。 メールの署名検証は、メール受信 PC のメール管理部の管理下で実行される。
B.09	活性化状態の署名プライベート鍵と署名公開鍵は、公開鍵証明書の有効期間が終了すると非活性化状態へ遷移する。
B.10	署名プライベート鍵や署名公開鍵は、公開鍵証明書の有効期間が終了すると OS の機能及びメールアプリケーションの機能で、送信メールに署名ができなくなり、メールの署名検証も有効期限切れ表示になり、非活性化状態へ自動的に遷移する。
B.11	非活性化日時の変更は不可なので、対象外。
B.12	公開鍵証明書の有効期間が終了する一月前に、情報システム部の公開鍵証明書発行依頼の担当者が、メール送信者に証明書の有効期間の終了が近づいたので、新たに署名プライベート鍵と署名公開鍵を生成し、証明書を再申請するように通知する。
B.13	● 以下の状況のいずれかが発生した場合は失効処理を行う。 ➢ PC が保護されない状況で PC 利用者以外のエンティティがアクセス可能であった場合

	➢ 署名プライベート鍵に不正なアクセスがあったことを示すログが検出された場合 ➢ メール送信者が認識していない署名付きメールの送信ログが確認された場合 ➢ その他、署名プライベート鍵の紛失・漏えいが疑われる事象が発生した場合 ● 上記の事象が生じた場合、該当する署名プライベート鍵の所有者は情報システム部に連絡し、情報システム部は当該署名プライベート鍵に対応する署名公開鍵の公開鍵証明書の失効処理を CA に依頼する。PC 利用者は当該署名プライベート鍵の使用を停止する。 ● メール受信者は B.14 の方法により配布される証明書失効リストを受信したら、OS の機能により該当する証明書を失効させる。
B.14	公開鍵証明書の失効を依頼された CA は、当該証明書に対する証明書失効リストを作成する。 通知は、CA から得た証明書失効リストを関係するエンティティに配布することにより行う。また、証明書に記載された OCSP にアクセスすることで失効情報を得ることも可能である。
B.15	B.02 の通り、一時停止状態への遷移は設けないため、対象外。
B.16	同上
B.17	同上
B.18	同上
B.19	同上
B.20	署名プライベート鍵・署名公開鍵が非活性化状態になって一定期間が経過し鍵を必要とするケースはないと判断したとき、または危殆化状態の署名プライベート鍵・署名公開鍵について危殆化状態での管理を行う必要がないと判断したとき、署名プライベート鍵を OS の機能を使用して当該鍵の所有者が手動で削除することで当該鍵の破壊を行う。署名プライベート鍵の破壊は局所的（Local）なものである。 署名公開鍵については公開鍵証明書にて管理されており、公開鍵証明書の有効期間の経過後、OS の証明書管理機能により自動的に削除され、破壊される。署名公開鍵の破壊は局所的（Local）なものである。
B.21	鍵の破壊は B.20 で慎重に判断するので、通知は必要としないため、対象外。
B.22	署名プライベート鍵及び署名公開鍵の鍵生成は FIPS 186-4 で規定されている方法で生成する。
B.23	乱数生成器は SP 800-90 で規定されている方法を使用し、乱数生成器で使用するエントロピーは Linux の乱数生成の疑似デバイスである/dev/random から得る。
B.24	鍵導出機能や鍵更新機能は使用していないため、対象外。
B.25	同上
B.26	対称鍵は利用しないため、対象外。
B.27	● CA から公開鍵証明書を受信したとき、受信した PC で、署名公開鍵が FIPS 186-4 で規定されたドメインパラメタを使用していることを検証する。

	● 署名付きメールを受信したとき、受信した PC で、一緒に受信した公開鍵証明書に記載された署名公開鍵が FIPS 186-4 で規定されたドメインパラメタを使用していることを検証する。
B.28	● CA から公開鍵証明書を受信したとき、受信した PC で、その証明書に記載された署名公開鍵が、生成した署名プライベート鍵とペアになる公開鍵であることを検証する。 ● 署名付きメールを受信したとき、受信した PC で、一緒に受信した公開鍵証明書を検証することにより、公開鍵を検証する。
B.29	● CA から公開鍵証明書を受信したとき、受信した PC で、当該証明書について OS の信頼できる証明書パスを検証する。 ● 署名付きメールを受信したとき、受信した PC で、一緒に受信した公開鍵証明書について OS の信頼できる証明書パスを検証する。 ● 証明書失効リストを受信したときに、受信した PC で、当該リストを検証するために必要となる公開鍵証明書について OS の信頼できる証明書パスを検証する。
B.30	トラストアンカー管理機能は OS の証明書管理機能を使用する。
B.31	トラストアンカーの配布は独自に行わない。OS に標準で保存されているトラストアンカーを使用する。トラストアンカーは信頼できる OS の更新機能により自動更新する。
B.32	トラストアンカーは信頼できる OS の更新機能により、自動的に追加、変更、削除を行う。緊急にトラストアンカーを修正したい場合、システム管理者が管理者権限により、OS の証明書管理機能で追加、変更、削除を行う。一般の利用者による追加、変更、削除はできない設定にする。
B.33	公開鍵の有効期間の延長は認めないため、対象外。
B.34	同上
B.35	メール送信者の CSR は組織内の情報システム部の公開鍵証明書発行依頼の担当者が所有者情報などを目視確認後、情報システム部から CA に CSR を送信し、CA から受信した公開鍵証明書は情報システム部が受信し、公開鍵証明書発行依頼の担当者が確認後にメール送信者に送られる。
B.36	メール受信者は、メールの署名を検証することで、一緒に受信した公開鍵証明書に記載された公開鍵とペアになる署名プライベート鍵をメール送信者が所持していることを検証する。
B.37	新規公開鍵証明書の発行のために情報システム部の公開鍵証明書発行依頼の担当者が CSR を受け取ったときに、その担当者は CSR に保存されたメタデータと署名を確認し、CSR 作成者がプライベート鍵の所有者であることを確認する。
B.38	鍵の有効期間、所有者情報（Subject）、key usage などのメタデータが、署名公開鍵に関連付けられ、署名公開鍵と一緒に公開鍵証明書に記載される。これらのメタデータと署名プライベート鍵とは、対応する署名公開鍵を介して関連付けられる。
B.39	署名公開鍵に関連付けられたメタデータは、公開鍵証明書に記載され、CA の署名により完全性が保護される。署名プライベート鍵は、ペアとなる署名公開鍵の保証により、完全性の関連付けが保証される。

B.40	メタデータの変更は認めないため、対象外。
B.41	メタデータの削除は認めないため、対象外。
B.42	同上
B.43	鍵メタデータのリスト化は認めないため、対象外。

2.4　鍵情報の保管方法

①　保管中の鍵情報のセキュリティを確保するための手段の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.44	FR6.73	CKMS 設計は、鍵情報（暗号鍵やメタデータ）をストレージに入れるエンティティの ID 認証及び認可検証に使用される手段を明記しなければならない。	6.5 節
B.45	FR6.74	CKMS 設計は、ストレージに入力する鍵情報（暗号鍵やメタデータ）の完全性検証に使用される手段を明記しなければならない。	6.5 節
B.46	FR6.75	CKMS 設計は、保管された対称鍵、プライベート鍵及びメタデータの機密性保護に使用される手段を明記しなければならない。	6.5 節
B.47	FR6.76	鍵ラッピング鍵（又は鍵ペア）が保管された鍵を保護するために使用される場合、CKMS 設計は、鍵ラッピング鍵（又は鍵ペア）を保護し、その使用を制御するために使用される手段を明記しなければならない。	6.5 節
B.48	FR6.77	CKMS 設計は、保管された鍵情報（暗号鍵及びメタデータ）の完全性保護に使用される手段を明記しなければならない。	6.5 節
B.49	FR6.78	CKMS 設計は、保管された鍵へのアクセスがどのように制御されるかを明記しなければならない。	6.5 節
B.50	FR6.79	CKMS 設計は、全ての保管された鍵を訂正又は復元するために使用される手法を明記しなければならない。	6.5 節

解説・考慮点

鍵情報の保管にあたっては、全ての暗号鍵は完全性保護を、加えて対称鍵及びプライベート鍵は機密性保護も必要とする。
また、認可されたユーザのみが保管された鍵情報にアクセスできるようにすべきであり、そのためのアクセスコントロールが必要である。
CKMS の設計にあたって、項目 B.44 及び B.45 は鍵情報を保管する時点での完全性保護を実現するための要求事項を、B.46～B.49 は保管中の鍵情報の完全性及び機密性の保護を実現するための要求事項を明確化することを求めたものである。B.50 は、保管中の鍵情報が破損した際

の対策を明確化することを要求したものである。

暗号アルゴリズムを安全に使う上で、(i) 利用権限を有する正しいエンティティ（利用者）のみが暗号鍵を利用できること、(ii) 暗号鍵の完全性が確保されていること、(iii) 加えて対称鍵及びプライベート鍵は機密性が確保されていること、は絶対条件である。また、暗号鍵の可用性の観点では、(iv) 利用権限を有する正しいエンティティ（利用者）は、暗号鍵の有効期間内はいつでも当該暗号鍵が利用できること、が求められる。

したがって、暗号鍵の保管にあたって、上記の要件を満たすように、鍵情報のセキュリティを確保するための手段を決める必要がある。なお、ここで決める手段は、後述の②以降で利用する手法を実現するものでなければならないことに留意されたい。

a) 当該暗号鍵をストレージなどに保管する時：

暗号鍵を保管する時点で、「偽物の暗号鍵」が正しく保管されてしまうと、その後の保護手段がいくら強固で正しく動作したとしても全く意味をなさない。そのようなことが起きないようにするには、「格納権限がある信頼できるエンティティ（利用者）」によって「正しい暗号鍵」が「正しく保管」されることが重要である。つまり、「格納権限がある信頼できるエンティティ(利用者)」であることを確認するための認証認可機能、「正しい暗号鍵」であることを確認するための完全性検証機能、さらに対称鍵やプライベート鍵では秘密裏に「正しく保管」するための機密性保護機能が求められる。これらの機能を実現する具体的な手段を決めるよう求めたものが検討課題 B.44～B.46 に該当する。

検討課題 B.47 は、暗号鍵の機密性保護のために当該暗号鍵の再暗号化を行う場合、再暗号化で利用する暗号鍵（鍵ラッピング鍵）も安全に管理・運用されていることが必要となるので、そのための必要となる機能を実現する具体的な手段を決めるよう求めている。

b) ストレージなどに保管された当該暗号鍵を利用する時：

(i) の要件を満たすためには、利用権限を有する正しいエンティティ（利用者）のみが保管された暗号鍵にアクセスできることを保証するためのアクセス制御機能・認証認可機能が求められる。この機能を実現する具体的な手段を決めるよう求めたものが検討課題 B.49 となる。

また、ストレージなどでの保管中も、誤操作や故障、改ざん攻撃などによって、暗号鍵の完全性が損なわれないように保護することが必要である。例えば、誤り検知や攻撃検知などによって予期せぬ理由で暗号鍵が書き変わらないようにする、利用前には読み出した暗号鍵の完全性の検証を行う、などの対策が考えられる。このような、保管中の鍵情報の完全性保護機能を実現する具体的な手段を決めるよう求めたものが検討課題 B.48 である。

検討課題 B.50 は、何らかの理由で保管された暗号鍵の完全性が損なわれた場合に、当該暗号鍵を正しい状態に復旧するための取られる方法を具体的に決めるよう求めたものである。例えば、誤りが検知された暗号鍵について訂正可能であれば適切な訂正を行い、訂正不能な場合には当該暗号鍵はしないようにするとか、暗号鍵のバックアップやアーカイブを行い、必要に応じて暗号鍵の復元に行うなどが考えられる。

最近はクラウドサービスを利用するケースも増えており、通常、クラウド上にあるデータに対する暗号処理もクラウド内で行われる。その際に利用する暗号鍵の管理・保管方法として、クラウド事業者に完全に任せる方式からクラウド利用者が自ら管理する方式までいくつかのやり方がある。代表的な方法として、クラウド事業者が全ての鍵管理を行う方式（Cloud Native Encryption Services）、鍵生成と管理はクラウド利用者が行うが暗号鍵の保管はクラウド事業者で行う方式（BYOK：Bring Your Own Key[9]）、クラウド利用者が全ての鍵管理を行う方式（HYOK：Hold Your Own Key[10]）などがある。

これらの方式のうち、どのような方式を取るかによって暗号鍵に対するクラウド利用者の管理度合いやクラウド事業者に求めるセキュリティレベルが変わる。一般に、クラウド事業者に鍵管理を依存する方式になるほど暗号鍵管理に関する負荷を低減させることができる一方、その安全性はクラウド事業者に依存するようになる。反対に、クラウド利用者が自ら鍵管理を行う方式になるほど、安全性をクラウド事業者に依存しなくてすむようになるが、オンプレミスで暗号鍵を管理するのと同様の負荷が求められるようになる。

これらの詳細については、日本クラウドセキュリティアライアンスが発行している Cloud Data Protection[11]を参照されたい。

項目 B.44～B.50 は、暗号鍵の管理を CKMS 運用者が自ら実施することを前提としたときの暗号鍵の保管方法について CKMS 設計で考慮すべき項目となっている。したがって、クラウドサービスを利用する場合には、どの部分の暗号鍵の管理をクラウド事業者に任せ、どの部分をクラウド利用者自ら管理するのかを切り分けること（暗号鍵管理についての責任分界点を明確化すること）が最初にすることである。そのうえで、クラウド利用者自ら管理する必要がある部分については他と同様に検討し、クラウド事業者に任せる部分についてはクラウド事業者が提供する機能や利用するサービス内容などをわかる範囲で記載すればよい。

例えば、クラウド事業者が全ての鍵管理を行う方式であれば、B.44 や B.49 のアクセス制御に係る部分を重点的に検討し、利用権限を有する正しいエンティティ（利用者）のみが暗号鍵を使えるようにすればよい。その他の項目については、クラウド事業者に管理を委ねる部分になる。

一方、BYOK の場合は、B.44 と B.49 に加え、クラウド側に暗号鍵を移す際の方法（B.45）や暗号鍵の復元方法（B.50）についても検討することが必要となる。

---

9 利用者が暗号鍵を作成してクラウドサービスの CKMS に持ち込む方式

10 利用者が管理する CKMS をクラウドサービスが利用する方式。オンプレミスで実現する場合やクラウド事業者が機能提供する場合、その併用など、いろいろな実現形態がある。Microsoft Azure Double Key Encryption、Google Cloud External Key Manager、Salesforce Cache-Only Key、Microsoft Azure Dedicated HSM、AWS CloudHSM など

11 日本クラウドセキュリティアライアンス、Cloud Data Protection、https://www.cloudsecurityalliance.jp/site/wp-content/uploads/2021/08/Cloud_Data_Protection2_V10.pdf

②　運用中の鍵情報の保管場所及び保護方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.51	FR6.46	それぞれの鍵タイプに対して、CKMS 設計は、以下のことを明記しなければならない：それぞれの鍵タイプとそのメタデータが保管される状況、鍵とメタデータの保管場所、及び鍵とメタデータの保護方法。	6.4.14 節

解説・考慮点

運用中の鍵情報がどこに存在し、どのような保護状態に置かれているのかを全て明らかにしておく必要がある。
項目 B.51 は、CKMS の設計にあたって、鍵情報の保管場所や保護方法などの要求事項を明確化することを求めたものである。運用中の鍵情報は、B.51 で定めた保管場所以外に置かれないようにしなければならない。

運用中の暗号鍵は、ストレージなどに保管された状態から読み込まれ、メモリ上などに保存された状態で暗号処理に使われるのが一般的である。その際、暗号処理のスループットを上げるため、暗号鍵が平文の形で置かれることが多く、暗号鍵の漏えいリスクが高い場所の一つになっている。

そのため、暗号鍵の漏えいリスクの低減策として、運用中の暗号鍵がどこに存在し、どのように保護されているのかを把握しておくことが重要であり、項目 B.51 はそれらの情報を具体的に決めるよう求めたものである。例えば、一言に「メモリ上に保存される」といっても、他のアプリケーションなどからも読み出せる汎用のメモリ上に置かれるのか、外部からの侵入が厳しく制限される暗号モジュール内のメモリ上に置かれるのかによって、その暗号鍵が置かれている保護状態は全く異なる。具体的には、前者は OS レベルでのアクセス制御による保護であるのに対して、後者は暗号モジュールが提供するアクセス制御のほか、当該暗号モジュールへの侵入検知による物理的防護などの保護が受けられる可能性がある。

なお、クラウドサービスでの運用中の鍵情報の管理方法について検討する必要があるかどうかは、暗号鍵の保管方法と同様、暗号鍵を管理する主体がクラウド事業者なのかクラウド利用者なのかによって異なる。つまり、前者の場合、運用中の暗号鍵を安全に管理するのはクラウド事業者の責任となるので項目 B.51 は対象外としてよい。一方、後者の場合は、運用中の暗号鍵を安全に管理するのはクラウド利用者の責任となるので B.51 も検討しておく必要がある。

③　鍵情報のバックアップ方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.52	FR6.47	CKMS 設計は、どのように、どこで、どのような状況において鍵及びそのメタデータがバックアップされるかを明記しなければならない。	6.4.15 節
B.53	FR6.48	CKMS 設計は、バックアップされた鍵情報（暗号鍵及びメタデータ）の保護のためのセキュリティポリシーを明記しなければならない。	6.4.15 節
B.54	FR6.49	CKMS 設計は、鍵情報（暗号鍵及びメタデータ）のバックアップ中のセキュリティポリシーがどのように実装されるかを明記しなければならない。例えば、バックアップされた鍵情報（暗号鍵及びメタデータ）の配送及び保管中における、機密性とマルチパーティコントロールの要求事項の実装方法。	6.4.15 節

解説・考慮点

CKMS の設計にあたって、項目 B.52 は、鍵情報のバックアップを行うための条件を明確化することを、B.53 及び B.54 はバックアップされる鍵情報のセキュリティ、特に機密性保護を確保するための要求事項を明確化することを求めたものである。
なお、鍵情報のバックアップを実施しない場合には、検討対象外である。

運用中の暗号鍵が喪失、改変、又はその他の理由で利用不能状態になったときに、当該暗号鍵を復元できるようにするため、安全な設備・メディアなどにバックアップをする場合がある。バックアップをすることにより、暗号鍵が喪失、改変又はその他の理由で利用不能になった場合であっても、当該暗号鍵で保護されている情報まで喪失する事態を避けることが可能になる。一方、暗号鍵の複製を作ることでもあるので、暗号鍵の漏えいリスクを高めることにつながる。

したがって、ここでの項目を検討する前に、まずはバックアップを行うことによるメリットとデメリットを天秤にかけて、暗号鍵のバックアップを行うかどうかを決定することが重要である。もしバックアップは行わないと決定した場合には、ここでの項目 B.52～B.54 は対象外となる。

一方、バックアップを行うと決定した場合には、できる限り、暗号鍵の複製を作ることによる漏えいリスクを低減する対策を検討する必要があり、そのためには、項目 B.52～B.54 の内容に沿って具体的なバックアップの条件や実施方法などを取りまとめておくことが重要である。特に、B.52 はバックアップを実施する条件を、B.53 はバックアップされる暗号鍵の保護方針を明確化することを求めており、バックアップを行う上での全体方針を決めるものである。この方針によって、バックアップのセキュリティが決まると言っても過言ではない。

B.54 は、B.53 の方針に沿った具体的なバックアップの実現方法を明確化することを求めている。

なお、B.52～B.54 では明確には求められていないが、バックアップした暗号鍵が使用されるこ

とがなくなったとき、当該暗号鍵は復元できないように破壊されるべきである。バックアップストレージメディアに保管されている場合にはメディア内の暗号鍵を、コピーが存在する場合には当該コピーも含めて破壊することを、B.53 でのセキュリティポリシーの中に明記することを強く推奨する。

④　鍵情報のアーカイブ方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.55	FR6.50	CKMS 設計は、どのように、どこで、どのような状況で鍵情報（暗号鍵やメタデータ）がアーカイブされるかを明記しなければならない。	6.4.16 節
B.56	FR6.51	CKMS 設計は、鍵情報（暗号鍵やメタデータ）のセキュアな破壊、又は新しい保管メディアに書き込まれた後の古い保管メディアのセキュアな破壊のための手法を明記しなければならない。	6.4.16 節
B.57	FR6.52	CKMS 設計は、アーカイブ鍵の暗号鍵有効期間（cryptoperiod）の期限切れ後に、鍵情報（暗号鍵やメタデータ）がどのように保護されるかを明記しなければならない。	6.4.16 節

解説・考慮点

CKMS の設計にあたって、項目 B.55 は、鍵情報のアーカイブを行うための条件を明確化することを、B.56 はアーカイブされた鍵情報を破棄するための要求事項を明確化することを、B.57 は機密性保護の継続性を確保するための要求事項を明確化することを求めたものである。
なお、鍵情報のアーカイブを実施しない場合には、検討対象外である。

バックアップが、「運用中の暗号鍵」が喪失、改変又はその他の理由で利用不能状態になった時の「復元対策」であるのに対して、アーカイブは、「運用中の暗号鍵かどうかに関わりなく」、法律や規則等で要求される期間中、必要に応じて、当該暗号鍵を復元できるようにしておく「長期保管対策」である。

バックアップもアーカイブも、暗号鍵の複製を作ることで漏えいリスクを高めることにつながるという点では同じであるが、バックアップが比較的頻繁に復元を行うことが想定されているのに対して、アーカイブは長期保管用ストレージ設備などに保管され、限定的な条件下でのみ復元されることが想定されている点が異なる。

アーカイブは、適用される法律や規則等も考慮して最小限の範囲で実施し、暗号鍵の複製を作ることによる漏えいリスクを低減する対策を検討する必要がある。

なお、バックアップと比較して、復元頻度が少なく、長期保管が想定されることから、可用性よりも機密性保護を優先して対策を考えるべきである。例えば、鍵分割による保護、アーカイブ鍵による暗号鍵の再暗号化、物理的に保護される装置内での保管、などがある。また、アーカイ

ブする必要がなくなれば、当該暗号鍵が復元できないようにそのアーカイブは破壊されるべきである。

項目 B.55 は、暗号鍵のアーカイブを行うための条件や保護方針を明確化することである。

鍵情報のアーカイブを実施しない場合には、ここでの項目 B.55～B.57 は対象外となる。一方、アーカイブを実施する場合には、アーカイブされた暗号鍵は、アーカイブされている期間、物理的又は暗号学的に保護されなければならない。また、保管メディアにアーカイブする作業中に、アーカイブする暗号鍵が露見しないように対策を取る必要がある。これらの方針を明確化することにより、アーカイブを行う上での全体方針を決めるものである。この方針によって、アーカイブのセキュリティが決まると言っても過言ではない。

B.56 は、アーカイブされた暗号鍵、もしくは当該暗号鍵がアーカイブされている保存媒体そのものを破壊するための要求事項を明確化することを求めたものである。また、B.57 は、アーカイブ鍵による暗号鍵の再暗号化によって機密性保護を行っているケースにおいて、アーカイブ鍵のほうが先に有効期間切れになった時に、機密性保護の継続性を確保するための要求事項を明確化することを求めたものである。例えば、アーカイブ鍵の暗号鍵有効期間が期限切れになる前の再暗号化や、新しいセキュアな保存メディアへの移動などである。

⑤　鍵情報の復元方法の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.58	FR6.53	CKMS 設計は、鍵情報（暗号鍵やメタデータ）の CKMS 復元ポリシーを明記しなければならない。	6.4.17 節
B.59	FR6.54	CKMS 設計は、鍵情報（暗号鍵やメタデータ）の復元ポリシーを実装及び実行するために使用されるメカニズムを明記しなければならない。	6.4.17 節
B.60	FR6.55	CKMS 設計は、どのように、どのような状況で鍵情報（暗号鍵やメタデータ）がそれぞれの鍵データベース又はメタデータ保管設備から復元されるかを明記しなければならない。	6.4.17 節
B.61	FR6.56	CKMS 設計は、鍵情報（暗号鍵やメタデータ）が復元中にどのように保護されるかを明記しなければならない。	6.4.17 節

解説・考慮点

CKMS の設計にあたって、項目 B.58～B.61 は、バックアップやアーカイブされた鍵情報を復元するための条件や復元方法、要求事項を明確化することを求めたものである。

バックアップやアーカイブされている暗号鍵を復元することは、当該暗号鍵のコピーを利用可能にする行為であるから、万が一にも不適正に復元が行われれば、即、当該暗号鍵の漏えいにつ

ながるほど危険な行為となる。したがって、バックアップやアーカイブからの復元が不適正に行われることがないように、厳格な復元ルールを規定し、そのルールが全て満たされていることを検証された後に、認可されたエンティティ（利用者）によって復元できるようにすべきである。

項目 B.58 は、復元ポリシーとして復元を実行するために必要なルールを定めることを求めている。B.59 はそのルールを守られるように具体的にどのように実装し、実行するのかを具体化すること、B.60 は復元処理を行う際の保管メディアへのアクセス条件を明確化することを求めている。また、B.61 は、予定外の場所で暗号鍵が露見しないような対策を求めたものである。

なお、ここで実現されるセキュリティ水準は、バックアップやアーカイブで実現されるセキュリティ水準と整合的であることが重要である。また、バックアップとアーカイブのどちらも実施しない場合には、本項目 B.58～B.61 は対象外である。

《トイモデルと記載例》

本節のトイモデルも、2.1 節のトイモデルと同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節のトイモデルで定めたものとする。

なお、このトイモデルでは、暗号鍵が喪失、改変、又はその他の理由で利用不能状態になったとしても影響範囲が限定的（署名生成ができなくなるだけでその他に影響しない）であるため、鍵情報のバックアップとアーカイブのどちらも実施しないこととする。鍵情報の喪失や破損が発生したときは、新たな暗号鍵を再生成し公開鍵証明書を発行し直す。このため、バックアップとアーカイブに関連する項目は対象外となる。

また、暗号鍵の保管に関する運用条件を以下のように設定する。【】内は記載例のどの項目に影響を与えているのかを示している。

- 鍵情報の機密性保護や完全性保護は、OS のアクセスコントロールシステム（ACS: Microsoft Windows の ACL、Linux のパーミッション機能）により保護する。【B.44, B.46, B.48】
- ストレージに入力する鍵情報において、署名プライベート鍵は証明書の公開鍵とペアになっていることを確認し、メタデータは証明書の内容と一致することを確認する。【B.45】
- OS の ACS により、基本的に鍵を生成したユーザ以外、鍵情報にアクセスできない。【B.49, B.51】
- 暗号鍵のバックアップとアーカイブは実施しない。【B.52～B.61】
- 保管していた鍵が使用できなくなった場合の復元手段は用意しない。【B.50】
- 保管された鍵を保護するために鍵ラッピング鍵や鍵ペアは使用しない。【B.47】

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.44	PC にログイン ID 及びパスワードを設定し、その ID 及びパスワードによる利用者認証を通過したエンティティに対してのみ、OS のアクセスコントロールシステム（ACS: Microsoft Windows の ACL、Linux のパーミッション機能）による自分が管理する鍵情報へのアクセス権限を付与する。 なお、システム管理者は、管理者 ID 及び管理者パスワードによる利用者認証でログインした場合に限り、例外的に全ての利用者の鍵情報に対してアクセス権限（Microsoft Windows の管理者権限、Linux の root 権限）が付与される。
B.45	ストレージに入力する署名プライベート鍵は、公開鍵証明書に記載された公開鍵とペアになっていることを確認し､当該証明書の CA 署名を検証することで完全性を検証する。
B.46	署名プライベート鍵と関連するメタデータは B.44 に記載した ACS により機密性を保護する。
B.47	保管された鍵を保護するために鍵ラッピング鍵や鍵ペアを使用しないため、対象外。
B.48	署名プライベート鍵と関連するメタデータは B.44 に記載した ACS により完全性を保護する。
B.49	署名プライベート鍵は B.44 に記載した利用者認証及び ACS を使用することで、システム管理者を除き、当該鍵を作成したユーザ以外はアクセスできない。
B.50	署名プライベート鍵の訂正や復元するための手段は用意しないため、対象外。
B.51	署名プライベート鍵、署名公開鍵や CA 署名公開鍵は署名するユーザの PC の内臓ストレージに保管する。これらの鍵は B.44 に記載した ACS で保護される。また、署名や検証するときに一時的に当該 PC のメモリ上に置かれるが、これは OS のプロセス間メモリ保護機能により保護される。
B.52	鍵情報のバックアップは実施しないため、対象外。 署名プライベート鍵や署名公開鍵が利用不能になった場合、その鍵の失効処理を行う。また、必要に応じて、新たな鍵ペアを再生成し公開鍵証明書を発行する。
B.53	同上
B.54	同上
B.55	鍵情報のアーカイブは実施しないため、対象外。
B.56	同上
B.57	同上
B.58	鍵情報のバックアップとアーカイブのどちらも実施しないため、対象外。
B.59	同上
B.60	同上
B.61	同上

2.5　鍵情報の鍵確立方法

①　鍵確立機能の利用局面の特定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.62	FR6.57	CKMS 設計は、どのように、どのような状況で鍵及びそのメタデータが確立されるかを明記しなければならない。	6.4.18 節

解説・考慮点

項目 B.62 は、CKMS の設計にあたって、鍵確立機能を利用する状況を特定することを要求したものである。

鍵確立（key establishment）機能とは、2 つ又はそれ以上のエンティティ間で暗号鍵をセキュアに共有するプロセスのことであり、方法として以下の 2 つがある。

- 鍵配送（key transport）:
  一方のエンティティが共有する暗号鍵を生成し、当該暗号鍵及び（あれば）メタデータを他方のエンティティに配付する。
- 鍵合意（key agreement）:
  両方のエンティティが共有鍵を導出するために使用される情報を共有し、当該情報から暗号鍵を導出する。

鍵確立が行われるタイミングは、「盗聴」という手段－すなわち、鍵確立に係る正当なエンティティに検知されない手段－で第三者が暗号鍵を窃取できる唯一のタイミングである。したがって、鍵確立を行うとき、(i) 確立される暗号鍵が誤りなく、正しく共有されること（暗号鍵の完全性）が求められるだけではなく、(ii) 関係するエンティティが全員正当であることの確認と（iii）セキュアな通信路での通信による暗号鍵の機密性保護が極めて重要である。

項目 B.62 は、CKMS の設計にあたって鍵確立機能を利用する状況を特定することで、鍵確立がいつ、どのように行われるのかを把握し、以降の②～⑤についての検討を忘れないようにすることを目的としている。②～⑤はいずれも（i）～（iii）の目的を達成するために必要な要求事項に関連するものである。

なお、鍵情報の鍵確立を使う状況がなければ、2.5 節の全て、すなわち B.62～B.70 全てが検討対象外となる。

②　鍵配送における鍵情報のセキュリティを確保するための要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.63	FR6.80	CKMS 設計は、配送中の対称鍵及びプライベート鍵の機密性保護に使用される手段を明記しなければならない。	6.6.1 節
B.64	FR6.81	CKMS 設計は、配送された鍵の完全性保護に使用される手段、及びエラー検出後にどのように鍵が再構築又は置き換えられるのかを明記しなければならない。	6.6.1 節
B.65	FR6.82	CKMS 設計は、配送される鍵素材（keying material）の鍵受信者に、どのように鍵送信者の識別子（ID）が認証されるかを明記しなければならない。	6.6.1 節

解説・考慮点

CKMS の設計にあたって、項目 B.63 は鍵配送における機密性保護のための要求事項を、B.64 は完全性保護のための要求事項を、B.65 は鍵送信者を確認するための要求事項を明確化することを求めたものである。
なお、鍵情報の鍵配送を使う状況がなければ検討対象外である。

鍵情報の鍵配送には、郵便・宅配便などの物理的手段を使う場合と、ネットワークを介する電子的手段を使う場合がある。いずれの手段であっても、全ての暗号鍵は完全性保護を、加えて対称鍵及びプライベート鍵は機密性保護も必要とする。

機密性保護の観点からは、物理的保護が行われる手段か、対称鍵ラッピング鍵又はひとつ以上の非対称配送鍵ペアが関わる鍵配送手段が使用される。前者であれば信頼できる仲介者が必要であり、後者であれば当該ラッピング鍵や配送鍵が配送に関わるエンドエンティティによって保護されたうえで信頼できる鍵配送手段が必要となる。項目 B.63 はこれらの要件に対応するために利用する手段を明確化することを目的としたものである。

完全性保護の観点からは、暗号鍵の送信者が信頼できることと、暗号鍵に改ざんやエラーがないことが求められる。したがって、配送された暗号鍵の受信者に対して、期待する認可された鍵送信者から当該暗号鍵が来たことを保証できることが必要である。また、暗号鍵の完全性検証を実行し、訂正可能な破損が検出された場合には適切な訂正を行い、訂正不能な破損が検出された場合には使用前に新しい又は訂正された暗号鍵を再確立する必要がある。
項目 B.64 は暗号鍵に改ざんやエラーがないことを確認するために利用する手段を、B.65 は鍵送信者が正当であることを鍵受信者が確認できるための方法を、それぞれ明確化することを目的としたものである。

なお、鍵情報の鍵配送を使う状況がなければ、B.63～B.65 は検討対象外である。

③　鍵合意における鍵情報のセキュリティを確保するための要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.66	FR6.83	CKMS 設計は、CKMS にサポートされるそれぞれの鍵合意スキームを明記しなければならない。	6.6.2 節
B.67	FR6.84	CKMS 設計は、鍵合意に参加するそれぞれのエンティティがどのように認証されるかを明記しなければならない。	6.6.2 節

解説・考慮点

CKMS の設計にあたって、項目 B.66 は鍵合意プロセスの手法について明確化することを、B.67 はエンティティの認証方法について明確化することを要求したものである。
なお、鍵情報の鍵合意を使う状況がなければ検討対象外である。

セキュアな鍵合意プロセスを利用する場合、そのプロセスに関与するそれぞれのエンティティは合意鍵を導出するために使われるある種の情報を提供しあうことで自ら合意鍵を生成することができるが、当該プロセスに関与していないエンティティは提供しあっている情報全て得たとしても合意鍵を得ることができない。逆に言えば、不正なエンティティに合意鍵を窃取されないためには認可されていないエンティティが鍵合意プロセスに不正に入り込むことを防止することが絶対条件となる。そのため、典型的には、鍵合意プロセスに参加する各エンティティは他方のエンティティ識別子の保証を必要とする。

項目 B.66 は鍵合意プロセスの手法について明確化することでセキュアな鍵合意プロセスであることを把握するため、また B.67 はエンティティの認証方法について明確化することで不正なエンティティに入り込むのを防止することを目的としたものである。

なお、鍵情報の鍵合意を使う状況がなければ、B.66 と B.67 は検討対象外である。

④　鍵確認機能を利用するための要求事項

項目	FR 番号	Framework Requirements の内容	SP800-130
B.68	FR6.86	CKMS 設計は、それぞれの鍵確認が実行される状況を明記しなければならない。	6.6.3 節
B.69	FR6.85	CKMS 設計は、他方のエンティティと正しい鍵を確立したことを確認するために使用されるそれぞれの鍵確認手段を明記しなければならない。	6.6.3 節

解説・考慮点

CKMS の設計にあたって、項目 B.68 は鍵確認を行うための条件を明確化することを、B.69 は

鍵確認の手法について明確化することを求めたものである。
なお、鍵確立した鍵情報の鍵確認を行う状況がなければ検討対象外である。

鍵確認機能は、鍵確立機能で共有された暗号鍵について、それぞれのエンティティが、実際に他方のエンティティが正しい暗号鍵を確立したことの確認をするために使用する機能である。

実際の暗号処理で利用する前に共有された暗号鍵を鍵確認することにより、何らかの理由で誤った暗号鍵が生成されたり、暗号鍵がうまく共有できなかったりした場合であっても、実害が発生する前に当該暗号鍵の利用を止めることが可能となる。

項目 B.68 はどのような状況のときに鍵確認が行うのかを把握することを、B.69 は鍵確認の手法について明確化することを求めたものである。

なお、鍵確立した鍵情報の鍵確認を行う状況がなければ、B.68 と B.69 は検討対象外である。

⑤　利用する鍵確立プロトコルの決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.70	FR6.87	CKMS 設計は、鍵確立と保管の目的のために CKMS によって採用されている全てのプロトコルを明記しなければならない。	6.6.4 節

解説・考慮点

項目 B.70 は、CKMS の設計にあたって、利用する鍵確立プロトコルを全て明確化することを要求したものである。ここで、定められた以外の鍵確立プロトコルを利用してはならない。

鍵確立で利用する通信路はセキュアであることが求められるので、CKMS 設計ではセキュアなプロトコルであることが確認されたものだけを使うべきである。項目 B.70 は、このことを確認する目的で、利用する鍵確立プロトコルを全て明確化することを求めたものである。

セキュアな鍵確立プロトコル、あるいは鍵確立機能を含んだプロトコルとして、代表的なものとして以下のものある。

- Internet Key Exchange（IKE）
- Transport Layer Security（TLS）
- Kerberos
- Over-The-Air-Rekeying（OTAR）Key Management Messages
- Secure Shell（SSH）

なお、鍵確立プロトコル自体は相互接続性の要求が強く求められることもあることから、「必ずしもセキュアとは言えないが相互接続性を実現するために必要」とされる手順や暗号アルゴリズ

ムもデフォルトで選択できるようになっていることがある。しかし、これらの手順や暗号アルゴリズムは真に必要な場合を除いて利用するべきではないので、意図せずに誤って利用することがないよう、CKMS 設計の段階でデフォルトでは使えないように設定しておき、真に必要な場合には「例外として意図的に設定変更する」ようにすることが重要である。

鍵確立プロトコルでの設定については、安全性と相互接続性のバランスを踏まえた推奨の設定ガイダンスが公開されているものも多い。そのようなガイダンスでは、「相互接続性を実現するために含まれたセキュアとは言えない手順や暗号アルゴリズム」はデフォルトでは使えないようにするための設定オプションが記載されている。

したがって、それらのガイダンスが存在する場合にはその設定に従うことで、セキュアな鍵確立プロトコルを確保できる。例えば、「TLS 暗号設定ガイダンス」や「SP 800-57 Part 3」などを参照されたい。

《トイモデルと記載例》

2.1 節と同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節で定めたトイモデルでは、暗号鍵の鍵配送及び鍵合意は使用しないため、B.62～B.70 は対象外となる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.62	B.02 の通り、暗号鍵の鍵配送及び鍵合意は使用しないため、対象外。
B.63	同上
B.64	同上
B.65	同上
B.68	同上
B.69	同上
B.70	同上

2.6　鍵情報の喪失・破損時の BCP 対策

①　鍵情報の喪失・破損に対する BCP 対策の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.71	FR10.11	CKMS 設計は、暗号鍵及びそのメタデータをバックアップ及びアーカイブするための手続きを明記しなければならない。	10.7 節
B.72	FR10.12	CKMS 設計は、保管又は伝送された破損した鍵情報（暗号鍵及びメタデータ）を復元又は置き換えるための手続きを明記しなければならない。	10.7 節

解説・考慮点

CKMS の設計にあたって、項目 B.71 は、BCP 対策として必要な鍵情報のバックアップを行うための手続きや要求事項を明確化することを、B.72 は BCP 対策として復旧を行うための手続きや要求事項を明確化することを求めたものである。なお、これらの項目で定めたことは B.52～B.61 の上位規定として機能し、B.52～B.61 の内容が B.71 及び B.72 の内容に矛盾してはならない。

鍵情報（暗号鍵やメタデータ）が喪失又は破損した場合で、バックアップもアーカイブもされていなかった場合、当該暗号鍵で保護されているデータの喪失につながる可能性がある。

とりわけ重大な災害は、多数の運用中の鍵情報の喪失又は破損を一気に引き起こす可能性が高い。この場合の BCP 対策として、鍵情報のバックアップやアーカイブは有効な手段であり、鍵情報の正当な復元を行うことで保護されているデータの喪失を防止することができる。その他のよくあるケースとしては、エンティティ（利用者）の誤操作や過失などによって、当人の鍵情報が破損したり紛失したりする場合があり、必要に応じて、バックアップからの復旧が求められることがある。

一方、バックアップやアーカイブからは要求したエンティティに対して復元した鍵情報を提供することになることから、万が一にも、そのエンティティが正しいエンティティでなかったり、正当な権限を持っていなかったりした場合は、復元要求を拒絶しなければならない。

また、鍵情報の喪失・破損の原因が紛失や攻撃など人為的な要因に起因する場合は、特に鍵情報の外部への流出などが否定できず、結果として当該暗号鍵で暗号化されていたデータの危殆化につながる可能性がある。この場合には、バックアップから単に当該鍵情報の正当な復元を行うだけでは不十分であり、当該暗号鍵の利用停止や失効処理、潜在的なリスク評価、新しい暗号鍵への置き換え及びデータの再暗号化といった、0 節の対応を含む一連の BCP 対策が必要となる。

そのため､鍵情報の喪失や破損時の BCP を実現するためにどのような対策が必要かを検討し、その結果、鍵情報のバックアップやアーカイブを行うこととした場合には、バックアップやアーカイブの方針をまず定める必要がある。この方針を具体化するものとして、項目 B.71 では、BCP 対策として必要な鍵情報のバックアップを行うための手続きや要求事項の明確化を､B.72 は BCP 対策として復旧を行うための手続きや要求事項の明確化を求めている。なお、これらの項目で定めたことは B.52～B.61 の上位規定として機能することから、B.52～B.61 の内容は B.71 及び B.72 の内容に沿って設定されなければならず､また矛盾していないことを確認することが重要である。

《トイモデルと記載例》

2.1 節と同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節で定めたトイモデルにおいて、2.3 節で定めた運用条件を行った場合､BCP を実現するための鍵情報のバックアップとアーカイブのどちらも実施しないことになっている。また、鍵情報の喪失や破損が発生したときは新たな暗号鍵を再生成し公開鍵証明書

を発行し直す。

このようなトイモデルでは、鍵情報の喪失・破損時の BCP 対策が必要ないため、B.71 と B.72 は対象外となる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.71	B.02 の通り、鍵情報のバックアップとアーカイブのどちらも実施しないため、対象外。
B.72	同上

2.7　鍵情報の危殆化時の BCP 対策

①　暗号鍵の危殆化に対する BCP 対策の決定／②　メタデータの危殆化に対する BCP 対策の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.73	FR6.102	CKMS 設計は、システムによって使用されているそれぞれの鍵タイプの受け入れ可能な暗号鍵有効期間（cryptoperiod）又は利用制限（usage limit）の範囲を明記しなければならない。	6.8.1 節
B.74	FR6.103	それぞれの鍵に対し、CKMS 設計は、セキュリティがその鍵に依存する他の鍵タイプを明記しなければならず、また初期鍵の危殆化が発生した時にそれに依存する鍵がどのように置き換えられるかを明記しなければならない。	6.8.1 節
B.75	FR6.104	CKMS 設計は、鍵が危殆化したときに他の危殆化した鍵を特定できるための手段を明記しなければならない。例えば、鍵導出鍵が危殆化したとき、導出された鍵をどのように特定するのか？	6.8.1 節
B.76	FR6.105	導入されたそれぞれの鍵タイプに対して、CKMS 設計は、どのメタデータ要素が危殆化（機密性、完全性、又はソース）しやすいのかを明記しなければならない。	6.8.2 節
B.77	FR6.106	CKMS 設計は、鍵のそれぞれの危殆化しやすいメタデータ要素に危殆化（機密性、完全性、又はソース）が起こったときに、起こり得るセキュリティ結果を明記しなければならない。	6.8.2 節
B.78	FR6.107	CKMS 設計は、それぞれの危殆化しやすいメタデータ要素での危殆化からどのように回復できるかを明記しなければならない。	6.8.2 節

解説・考慮点

そもそも全ての潜在的なセキュリティ問題を CKMS が防止し鍵情報の危殆化が発生しないようにすることは現実的でないことを前提として、CKMS の設計においては鍵情報の危殆化を速

やかに検知できるようにすべきである。
鍵情報の危殆化が検知された場合、次のステップを参考に、適切な当事者に危殆化を警告し、望ましくない影響を軽減し、最後にセキュアな状態に復帰することが必要である。

a) その原因及び範囲を決定するために危殆化を評価
b) 鍵情報（暗号鍵やメタデータ）の露出を最小化するために危殆化軽減手段を実行
c) 危殆化の再発を防止するために適切な是正手段を実施
d) CKMS をセキュアな運用状態に復帰させる

＜暗号鍵の危殆化＞
CKMS の設計にあたって、項目 B.73 は暗号鍵に対する暗号鍵有効期間の設定や利用範囲の制限について明確化することを、B.74 は危殆化した暗号鍵（の鍵タイプ）だけでなく連鎖的に影響を受ける可能性がある別の暗号鍵（の鍵タイプ）を含めてどのような BCP 対策を行うかを明確化することを要求したものである。B.75 は、危殆化した暗号鍵から連鎖的に影響を受ける別の暗号鍵の特定方法の明確化を要求したものである。

＜メタデータの危殆化＞
CKMS の設計にあたって、項目 B.76 はメタデータの中でも危殆化が起こりやすい又は関連する暗号鍵の危殆化につながりやすいものがどれであるかを明確化することを、B.77 はメタデータに危殆化が起きた時にどのような影響が出るかを明確化することを要求したものである。B.78 はどのような BCP 対策を行うかを明確化することを要求したものである。

暗号鍵の危殆化の影響は鍵タイプ及び鍵の用途に依存し、以下の結果をもたらし得る。

- 機密性の喪失
- 完全性の喪失
- 認証の喪失
- 否認防止の喪失
- これらの喪失の組み合わせ

一般に、暗号鍵が危殆化した場合には、当該暗号鍵の利用を停止し、新しい暗号鍵に置き換えるとともに、すでに暗号処理（暗号化や署名生成）が行われた情報に対しては個別にその正当性の判断を行うことになる。これらは、危殆化状態への遷移に相当し、B.13 や B.14 で決められた手段が取られる。

しかしながら、危殆化した暗号鍵の使われ方によっては、当該暗号鍵で保護されたデータに対してだけでなく、当該暗号鍵が保護する他の多くの暗号鍵についても危殆化を連鎖的に引き起こす可能性があることに留意されたい。例えば、システムマスター鍵など、上位の暗号鍵が危殆化すればシステム全体に影響が及ぶ可能性があり、鍵ラッピング鍵が危殆化すれば当該鍵で暗号化された鍵情報に影響が及ぶ。このような場合、危殆化した暗号鍵の失効・置き換えはもとより、

当該暗号鍵に依存して影響を受ける他の暗号鍵も可能な限り速やかに失効・置き換えを行うべきである。なお、暗号鍵の置き換えに伴い、データの再暗号化が必要となる場合もある。

また、暗号鍵の生成や更新に鍵導出手段を使っている場合、鍵導出手段に入力する元の暗号鍵が危殆化すると、それ以降に導出される暗号鍵も全て危殆化している。したがって、このような場合には、鍵導出手段を使うのではなく、改めて新しい暗号鍵を独自に生成し直す必要がある。

そこで、項目 B.74 と B.75 は、暗号鍵が危殆化した場合にどの程度の他の暗号鍵に影響を与える可能性があるかを、暗号鍵ごとに把握しておき、さらに BCP 対策として影響を受ける可能性がある暗号鍵の更新方法までを決めておくことを求めている。具体的には、B.74 は、危殆化した暗号鍵から連鎖的に影響を受ける可能性がある別の暗号鍵を含めてどのような BCP 対策を行い、暗号鍵を使う処理を再開するのかを明確化することを、B.75 は危殆化した暗号鍵から派生して生成される別の暗号鍵の特定方法の明確化を要求したものである。なお、鍵導出機能などを使わない場合には B.75 は対象外である。

暗号鍵の危殆化の影響を小さくするために、使用するそれぞれの暗号鍵に対して適切な暗号鍵有効期間の設定や利用範囲の制限をすることで、暗号鍵の危殆化のリスクを低減することも重要である。一般的には、対称鍵ラッピング鍵、鍵配送鍵、及び鍵合意鍵の暗号鍵有効期間を実用的な最短期間にしておくことがよい。この他、鍵導出鍵とマスタ鍵も定期的に変更したほうがよい。項目 B.73 は、暗号鍵の鍵タイプごとに有効期限や利用範囲を具体的に定めることを求めており、その範囲内で該当する暗号鍵を生成・利用するようにすることで、暗号鍵の危殆化の影響を小さくすることを目的としている。

メタデータの危殆化は、メタデータ要素及びその使われ方に依存して、暗号鍵の危殆化や当該暗号鍵によって保護されるデータの危殆化につながる可能性がある。項目が B.76 と B.77 は、メタデータのうち、どのデータが危殆化しやすいのか、さらにそのデータの危殆化が起きたら関連する暗号鍵にどのようなことが起きうるのかを具体的に把握することを求めている。B.78 は、危殆化状態からの復旧方法についての具体化を求めており、例えば、暗号鍵自体は危殆化していないことが確認でき、危殆化したメタデータの内容だけを更新すれば、当該暗号鍵の利用を再開できるようなケースで利用することを想定している。

③　役員・従業員によるセキュリティ危殆化に対する BCP 対策の決定

項目	FR 番号	Framework Requirements の内容	SP800-130
B.79	FR6.117	CKMS 設計は、それぞれのサポートされる役割に提供される、あらゆる役員・従業員による危殆化の検知機能を明記しなければならない。	6.8.7 節
B.80	FR6.118	CKMS 設計は、それぞれのサポートされる役割に提供される、あらゆる役員・従業員による危殆化を最小化する機能を明記しなければならない。	6.8.7 節

B.81	FR6.119	CKMS 設計は、それぞれのサポートされる役割に提供される、CKMS 危殆化からの回復能力を明記しなければならない。	6.8.7 節

解説・考慮点

CKMS の設計にあたって、項目 B.79 及び B.80 は役員・従業員によるセキュリティ危殆化への事前対策としての要求事項を明確化することを求めたものである。B.81 は危殆化が検知された後にどのような BCP 対策を行うかを明確化することを要求したものである。

CKMS のセキュアな運用に責任のある人間が、与えられた権限を悪用し、自らそのセキュリティを危殆化させる場合がある。そのような事態への対策としては、基本的に権限必要最小限ルールの徹底が重要であり、必要な人に必要な権限しか与えない、権限を悪用していないかを監査する、操作ログを隠蔽できないようにする、といった事前対策が重要となる。

また、役員・従業員によるセキュリティ危殆化が発生した場合には、あらかじめ決められた情報セキュリティポリシー及び CKMS 機能に基づいて、以下のような回復手続きで対応・復旧することが重要である。さらに、再発防止策としてのセキュリティポリシーや運用規程等の改訂もあり得る。

- システムの完全なシャットダウン
- 新しい暗号鍵によるバックアップ設備及びシステムの活性化
- 起こり得るセキュリティ障害についての現在及び潜在的ユーザへの通知
- 危殆化した暗号鍵へのフラグ付け・失効処理

項目 B.80 は上記に示したような事前対策に関する方針を実現するための手段を具体的に示すことを、B.81 は危殆化が検知された後にどのような BCP 対策を行い、暗号鍵を使う処理を再開するかを明確化することを要求したものである。

《トイモデルと記載例》

本節のトイモデルも、2.1 節のトイモデルと同じ、メールの送信元認証を S/MIME の署名付きメールで実現するシステムであり、暗号鍵のライフサイクルは 2.2 節のトイモデルで定めたものとする。

また、システムの運用条件を以下のように設定する。【】内は記載例のどの項目に影響を与えているのかを示している。

- 署名プライベート鍵と署名公開鍵の有効期間は、公開鍵証明書の有効期間とする。【B.73】
- 署名プライベート鍵が危殆化すると署名公開鍵も同時に危殆化したとみなす。【B.74, B.75】
- 特に、危殆化しやすいと想定できるメタデータ要素はない。【B.76～B.78】
- 役割を分離して権限を必要な範囲にしている。【B.79】

- 個々のメール利用者が使用する PC において、使用している OS のログ保存機能により、署名プライベート鍵のアクセスログを管理する。アクセスログはシステム管理者しか確認できない。【B.79, B.80】
- 情報システム部の公開鍵証明書発行依頼の担当者の公開鍵証明書発行依頼ログと操作ログが PC 内に保存され、操作ログへのアクセスは情報システム部の管理者しかできない。【B.79, B.80】
- 鍵の危殆化が疑われるときは失効処理を行い、鍵を再生成し、証明書を再発行する。【B.81】

以上のトイモデルにおける記載例は、以下の「署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例」のようになる。

署名付き S/MIME を使用しメール送信元認証をするシステムにおける記載例

B.73	署名プライベート鍵と署名公開鍵の有効期間は、公開鍵証明書に記載された有効期間と一致する。システムの運用標準では 1 年プラス 10 日とする。
B.74	署名公開鍵は、署名プライベート鍵の危殆化に依存する。 署名プライベート鍵に危殆化の疑いが発生したときは、当該鍵ペアの両方の失効処理を行い、新たな鍵ペアを再生成し公開鍵証明書を発行し直すことにより、鍵の置き換えを行う。
B.75	署名プライベート鍵が危殆化すると署名公開鍵も同時に危殆化したとみなす。
B.76	公開鍵証明書に記載されているメタデータは、CA 署名により完全性が保護される。また、保存されるメタデータは B.44 に記載した ACS により完全性を保護されている。そのため、危殆化しやすいメタデータ要素は想定しない。
B.77	B.76 により、対象外。
B.78	同上
B.79	● 本システムでの役割として、以下の通りである。 ➢ メール利用者（メール送信者・メール受信者） ➢ 当該 PC ごとのシステム管理者（情報システム部内で PC ごとに担当者を割り当てる） ➢ 情報システム部の公開鍵証明書発行依頼の担当者 ➢ 情報システム部の公開鍵証明書発行依頼の管理者 ● メール利用者が使用する PC の署名プライベート鍵のアクセスログを保存する。システム管理者がアクセスログを毎月確認し、不正な署名生成が行われていないかどうかを確認する。 ● 公開鍵証明書発行依頼の担当者は、公開鍵証明書発行依頼ログを毎月確認し、不正な公開鍵証明書発行依頼が行われていないかどうかを確認する。 ● 公開鍵証明書発行依頼の管理者は、発行依頼の担当者の操作ログを毎月確認し、不正な公開鍵証明書発行依頼が行われていないかどうかを確認する。
B.80	● メール利用者が署名プライベート鍵を複製するなど、署名プライベート鍵に対するアクセスは全てアクセスログに記録される。

	● メール利用者は、署名プライベート鍵のアクセスログにアクセスできない。アクセスできるのは、システム管理者だけである。 ● PC は当該利用者が安全に管理する。 ● システム管理者は、メール利用者の許可又は別途規定に基づく手続きによる場合を除き、メール利用者の署名プライベート鍵にアクセスしてはならない。 ● 公開鍵証明書発行依頼の担当者が行う発行依頼は全て操作ログに記録される。 ● 公開鍵証明書発行依頼の担当者は、操作ログにアクセスできない。アクセスできるのは、公開鍵証明書発行依頼の管理者だけである。
B.81	● メール利用者が使用する PC のアクセスログにより署名プライベート鍵の危殆化を検知したシステム管理者は、CA に該当する鍵ペアの失効処理を依頼し、CA から得た証明書失効リストを関係するエンティティに配布する。 ● 公開鍵証明書発行依頼の管理者が、証明書発行依頼の処理に使用している PC の操作ログにより不正な証明書発行を検知した場合は、CA に該当する署名公開鍵の失効処理を依頼し、CA から得た証明書失効リストを関係するエンティティに配布する。 ● 失効処理した鍵ペアのメール利用者は新たな鍵ペアを再生成し、公開鍵証明書を新規発行する。

3 暗号アルゴリズムの選択

本章の目的・趣旨

本章は、設計指針（基本編）の 6 章に記載されている要求事項（各節での色付き枠内で示している内容）について解説したものである。

CKMS 設計では、要求される保護レベル（セキュリティ強度）を満たすように暗号アルゴリズムと鍵長を決定しなければならない。セキュリティ強度は、扱う情報の資産価値、求められる情報の機密性や完全性、保護する期間（保護終了年）を踏まえて決定すべきである。

決定したセキュリティ強度、暗号アルゴリズムと鍵長を明記することにより、どの程度安全に情報が保護されているかを確認することも可能になる。なお、本章で扱う暗号アルゴリズムとしては主に公開鍵暗号、デジタル署名、共通鍵暗号、ハッシュ関数を想定としているが、高機能暗号や秘密分散など、新しいタイプの暗号アルゴリズムを含めることもできる。

3.1 暗号アルゴリズムのセキュリティ

① 要求される保護レベル（セキュリティ強度）に対応した暗号アルゴリズムの決定

項目	FR 番号	Framework Requirements の内容	SP800-130
C.01	FR2.1	CKMS 設計は、システムによって使用される全ての暗号アルゴリズムとそれぞれのアルゴリズムでサポートされる全ての鍵長を明記しなければならない。	2.1 節
C.02	FR2.2	CKMS 設計は、鍵と鍵に結び付けられたメタデータを保護するために導入されているそれぞれの暗号技術について推定されるセキュリティ強度を明記しなければならない。	2.1 節

解説・考慮点

暗号アルゴリズムの選定方法について取り扱う。
CKMS がライフサイクル全体にわたって管理及び保護している暗号鍵を使用することで要求される保護レベル（セキュリティ強度）を満たすことができる暗号アルゴリズムを選定することが求められる。
項目 C.01 及び C.02 は、CKMS の設計にあたって、要求される保護レベル以上を実現していることを確認するために、採用している暗号アルゴリズム（鍵長を含む）及びセキュリティ強度の明確化を求めたものである。

本節で求めているのは、要求される保護レベル（セキュリティ強度）を決定し、それに対応した暗号アルゴリズム及び鍵長を選択し、明記することである。

具体的には、C.01 では、決定したセキュリティ強度に応じて使用可能な設定を行ったりして、当該システムで利用可能な全ての暗号アルゴリズムとサポートされる全ての鍵長を洗い出すことを求めている。また、C.02 では、C.01 に明記した暗号アルゴリズムで使われる暗号鍵とメタデ

ータを保護するために使用している暗号技術についても、どの程度のセキュリティ強度を有しているかを評価することを求めている。これは、C.01 で使う暗号アルゴリズムと鍵長が適切に選択されていたとしても、そこで利用する鍵情報が適切なセキュリティ強度で保護されていなければ、必要とされるセキュリティ強度が達成されないためである。

上記の要求を満たすためには、まず CKMS 設計では最初に必要なセキュリティ強度を決める必要がある。セキュリティ強度の決定では、扱う情報の資産価値、情報の機密性や完全性などのほか、該当システムの利用期間の終了年も重要な要因の一つとなる。なお、扱う情報によっては、当該情報の保護がシステムの終了年以降も必要な場合がある。例えば、攻撃者が暗号化された通信データを先に窃取しておいて解読が可能になった時期に復号を行う攻撃（Store (Harvest)-then-decrypt、Store now & decrypt later、Retrospective Decryption ともいう）を考慮する必要があるケースである。その場合、システムの終了年ではなく、当該情報の保護が必要な期間を基準にセキュリティ強度を決めるべきである。

具体的に必要なセキュリティ強度の決定にあたっては、「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準[12]」又は「暗号鍵設定ガイダンス[13]」を参考にされたい。

また、扱う情報の資産価値（重要性）では、扱う情報の機密性・完全性・可用性に危殆化が発生したときに、どの程度の影響を与えるかを段階的に評価し、その影響度が大きいほど資産価値が高いと判断される。この評価はリスク分析の一環として行われることが多く、資産価値が高い情報と評価されると、その情報を扱うシステムではより強いセキュリティが求められることがある。このような場合には、通常のセキュリティ強度よりも高い強度を設定することが同時に求められる場合もあることに留意されたい。必要があれば、「中小企業の情報セキュリティ対策ガイドライン[14]」や「政府機関等の対策基準策定のためのガイドライン（令和 3 年度版）改定版[15]」なども参考にされたい。

なお、情報資産は時間の経過により、陳腐化する、逆に価値が上がることもある。そのような情報資産の重要性の変化があるかどうかも含めて検討し、適切なセキュリティ強度を設定すべきである。

「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準」は、CRYPTREC 暗号リスト[16]に掲載されている暗号技術を利用する際に、適切なセキュリティ強度を実現するためのアルゴリズム及び鍵長の選択方法を規定したものである。したがって、利用する鍵長についてこの設定基準に合致しない鍵長を用いた場合には、電子政府推奨暗号リストの暗号技術を利用しているとは見なされないことに留意されたい。また、政府機関等のサイバーセキュリティ対策のための統一基準において適用対象となる電子政府システム（暗号化機能・電子署名機能の導入を行うものに限る。）の調達・開発・運用に関わる場合には、この設定基準に従う必要がある。

---

12 暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準、https://www.cryptrec.go.jp/list.html
13 暗号鍵設定ガイダンス、https://www.cryptrec.go.jp/op_guidelines.html
14 中小企業の情報セキュリティ対策ガイドライン第 3 版、
https://www.ipa.go.jp/security/guide/sme/about.html
15 政府機関等の対策基準策定のためのガイドライン（令和 3 年度版）改定版、
https://www.nisc.go.jp/pdf/policy/general/guider3_2.pdf
16 電子政府における調達のために参照すべき暗号リスト（CRYPTREC 暗号リスト）
https://www.cryptrec.go.jp/list.html

表 3-1 は、本設定基準において、システムの想定運用終了・廃棄年又は利用期間の終了年を基準に必要なセキュリティ強度要件を示したものである。例えば、128 ビットセキュリティは 2022～2040 年が利用可能であり、2041～2050 年が移行完遂期間になっているので、2050 年までに運用を停止するシステムであれば 128 ビットセキュリティでもよいが、2051 年以降も何らかの形で運用するシステムでは 2050 年までに 192 ビットセキュリティ以上のセキュリティ強度に移行することが必要となる。

表 3-1　セキュリティ強度要件の基本設定方針

<table>
<tr><th colspan="2">想定運用終了・廃棄年／利用期間</th><th>2022～2030</th><th>2031～2040</th><th>2041～2050</th><th>2051～2060</th><th>2061～2070</th></tr>
<tr><td rowspan="2">112 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">移行完遂
期間
((c)参照)</td><td>利用不可</td><td rowspan="2">利用不可</td><td rowspan="2">利用不可</td><td rowspan="2">利用不可</td></tr>
<tr><td>処理
((b)参照)</td><td>許容</td></tr>
<tr><td rowspan="2">128 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">移行完遂
期間
((c)参照)</td><td>利用不可</td><td rowspan="2">利用不可</td></tr>
<tr><td>処理
((b)参照)</td><td>許容</td></tr>
<tr><td rowspan="2">192 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td></tr>
<tr><td>処理
((b)参照)</td></tr>
<tr><td rowspan="2">256 ビット
セキュリティ</td><td>新規生成
((a)参照)</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td><td rowspan="2">利用可</td></tr>
<tr><td>処理
((b)参照)</td></tr>
</table>

(a) **新規に暗号保護を適用**する（例えば、暗号化や署名生成を実行する）際は、原則として、2040 年までは 128 ビット以上のセキュリティ強度のものを**選択すべきである**。2041 年以降は 192 ビット以上のセキュリティ強度のものを**選択すべきである**。

(b) **保護済みのデータに対して処理を実行**する（例えば、復号や署名検証を実行する）際は、2040 年までは 128 ビット以上、2041 年以降は 192 ビット以上のセキュリティ強度のものを**選択すべきである**。ただし、保護済みのデータに対する正当性を担保又は確認するための何らかの技術的又は運用的な対策やルール等（暗号技術によるものとは限らない）を併用している場合に、2031 年以降も 2040 年までの必要な範囲内で 112 ビットセキュリティ強度のものを選択することを許容する。同様に、2051 年以降も 2060 年までの必要な範囲内で 128 ビットセキュリティ強度のものを選択することを許容する。

(c) 移行完遂期間内に、よりセキュリティ強度の高い暗号技術又は鍵長への移行を完遂させることを前提として、利用する暗号処理が短期間で完結する場合（例：エンティティ認証）、又は既存の電子政府システムの継続利用やそれらとの互換性・相互接続性維持などの必要がある

場合には、2030 年までは 112 ビットセキュリティ強度のものを、2050 年までは 128 ビットセキュリティ強度のものを選択することを許容する。

「暗号鍵設定ガイダンス」は、安全な暗号技術の導入の観点から、暗号技術を利用する際の鍵長の選択方法に関する一般的な考え方を解説したものであり、必要な鍵長を判断する上での目安となるものである。表 3-2 は、本ガイダンスにおいて、1982 年当時に DES が有していたのと同程度のセキュリティ強度を実現するために必要と推定されるビットセキュリティを示したものであり、表 3-2 でのビットセキュリティを下限のセキュリティ強度として、一定のセキュリティマージン（数十ビット）を追加したそれ以上のセキュリティ強度で設定することが望ましいと記載されている。例えば、2040 年に該当システムの利用を終了するのであれば最低で 104 ビット以上、できれば 128 ビット以上のセキュリティ強度を設定するのがよい。

表 3-2　1982 年の DES と同等のセキュリティを提供すると推定される（＝その後 10～15 年程度安全と期待される）ビットセキュリティ

年	1982	2030	2040	2050	2060	2070
ANSSI（2014）	56	81 ～ 96	86 ～ 104	91 ～ 112	96 ～ 120	101 ～ 128
Lenstra（2001）	56	93	101	109	－	－
Lenstra（2004）	56	88	95	102	－	－

必要なセキュリティ強度を決定したら、暗号アルゴリズムと鍵長を決定する。その際、使用する全ての暗号アルゴリズムと鍵長が必要とするセキュリティ強度を上回る強度となるように選択しなければならない。公開鍵暗号、デジタル署名、共通鍵暗号、ハッシュ関数については、CRYPTREC 暗号リストの電子政府推奨暗号リスト（又は推奨候補暗号リスト）に掲載されている暗号アルゴリズムから選択することを推奨する。また、セキュリティ強度と鍵長の関係は、「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準」又は「暗号鍵設定ガイダンス」の 2.2 節「暗号技術の推定セキュリティ強度表現－ビットセキュリティ」を参照する。

《トイモデルと記載例》

本節のトイモデルは、Web ブラウザをクライアントとするクライアントーサーバシステムである。暗号プロトコルとして TLS 通信を使用する。

そこで、このトイモデルでは、セキュリティ強度は「暗号強度要件（アルゴリズム及び鍵長選択）に関する設定基準」に従い、また TLS サーバでの暗号スイートの設定では「TLS 暗号設定ガイドライン[17]」の「高セキュリティ型での暗号スイート推奨設定（TLS1.3 限定）」に従って設定している。

まず、該当システムは 2023 年から 8 年間使用し 2031 年末破棄する予定である。この場合、表

[17] TLS 暗号設定ガイドライン、https://www.ipa.go.jp/security/crypto/guideline/ssl_crypt_config.html, https://www.cryptrec.go.jp/op_guidelines.html

3-1 により必要なセキュリティ強度は 128 ビットとなり、TLS1.3 の中で利用する暗号アルゴリズムでの鍵長は 128 ビットセキュリティを満たす鍵長を利用することになる。

TLS1.3 を利用する場合における高セキュリティ型の暗号スイートは、

TLS_AES_256_GCM_SHA384、TLS_CHACHA20_POLY1305_SHA256、
TLS_AES_128_GCM_SHA256、TLS_AES_128_CCM_SHA256、
TLS_AES_128_CCM_8_SHA256

であり、個々の暗号スイートの最後の SHA384/ SHA256 は鍵生成 HKDF で使用するハッシュ関数を示している。

なお、TLS 暗号設定ガイドラインにおける高セキュリティ型での推奨設定では、暗号アルゴリズム自体の安全性に問題がないものだけが選定されているので、暗号アルゴリズムそのものの選択に関しては気にする必要はない。

また、このサーバで使われるサーバ証明書は、セキュリティ強度が 128 ビット以上必要なので、ここでは署名方式として ECDSA P-256、証明書に使用するハッシュ関数として SHA-256 を使用する。

このトイモデルを対象とした場合、C.01 については「TLS 暗号設定ガイドライン」の高セキュリティ型での推奨設定に従った内容となる。C.02 については、鍵とメタデータを保護するための暗号技術としては、認証局が署名したサーバ証明書によって「サーバ公開鍵」と「有効期間」とが結び付けられているので、推定セキュリティ強度は認証局の署名で使われている暗号アルゴリズムと鍵長から導かれるセキュリティ強度である。

以上のトイモデルにおける記載例は、以下の「クライアントーサーバシステムにおける記載例」のようになる。

クライアントーサーバシステムにおける記載例

C.01	署名 ECDSA P-256 - 鍵長 256 ビット SHA-256 鍵合意 ECDHE P-256 - 鍵長 256 ビット X25519 - 鍵長 255 ビット データ暗号化／復号 AES-128-GCM - 鍵長 128 ビット AES-256-GCM - 鍵長 256 ビット AES-128-CCM - 鍵長 128 ビット AES-128-CCM-8 - 鍵長 128 ビット CHACHA20-POLY1305 - 鍵長 256 ビット

	鍵生成（HKDF） HMAC-SHA-256 HMAC-SHA-384
C.02	サーバ公開鍵とメタデータ（有効期間）の結び付きの保護は、認証局の署名により行っている。そこで利用している暗号技術は、ECDSA P-256 と SHA-256 であるので、セキュリティ強度は 128 ビットセキュリティである

4 暗号アルゴリズム運用に必要な鍵情報の管理

本章の目的・趣旨

本章は、設計指針（基本編）の 7 章に記載されている要求事項（各節での色付き枠内で示している内容）について解説したものである。

「鍵情報の管理」の主要な効果の 1 つは、該当システムで使用している暗号鍵について、鍵タイプとメタデータに応じて分類した上で保護方法などを明記し管理することにより、その暗号鍵が安全に管理されていることを明確にすることである。

暗号鍵の管理方法は、扱う情報の資産価値や対策コストを考慮して CKMS 設計者が決定することとなるが、利用用途や目的等に応じた鍵タイプごとに、その特徴に合わせた相応しい鍵管理を実施することが必要になる。また、鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。

このように分類して鍵情報を管理することで、CKMS 設計者が明示的に管理すべき暗号鍵を漏れなく洗い出し、それらの暗号鍵を安全に管理していることを明確にすることができる。また、ある暗号鍵の危殆化が疑われるときに、その暗号鍵の管理状況の確認や危殆化による影響リスクの判断等に利用することができ、さらに、危殆化が発生した場合に敏速な対応も可能になる。

本章では、暗号鍵の安全な管理を行うために必要な、鍵タイプ、鍵のメタデータの分類方法、またそれらの保護方針や記載方法を解説する。

4.1 鍵情報の種類

SP800-130 に記載されている鍵タイプとメタデータを以下のように解説している。CKMS 設計者は、明示的に管理すべき暗号鍵すべてについて、暗号鍵の利用用途や目的に応じて、それぞれの暗号鍵がどの鍵タイプに属するか、管理対象となる鍵情報が何であるかを分類することが必要になる。

暗号鍵は、以下の通り、特性と用途（＋オプション）に応じて分類され、これらの組み合わせで「鍵タイプ」が定義される。

特性	公開（Public）	一般に公開できる情報
	プライベート（Private）	一人のユーザのみが秘密に保持する情報
	対称（Symmetric）	送信者と受信者が共通して秘密に保持する情報

オプション	静的（Static）	長期的に固定した情報
	一時的（Ephemeral）	1つのセッションやトランザクションでのみ使われる情報

用途	データの暗号化／復号（Encryption/Decryption）
	鍵ラッピング（Key Wrapping）
	鍵配送（Key Transport）

	鍵合意（Key Agreement）
	署名（Signature）
	認証（Authentication）
	認可（Authorization）
	乱数生成（Random Number Generator（RNG））
	マスタ鍵（Master Key）

SP800-130 で分類する鍵タイプは以下の通りである。

1）署名プライベート鍵（Private Signature Key）
2）署名公開鍵（Public Signature Key）
3）認証対称鍵（Symmetric Authentication Key）
4）認証プライベート鍵（Private Authentication Key）
5）認証公開鍵（Public Authentication Key）
6）データ暗号化／復号対称鍵（Symmetric Data Encryption/Decryption Key）
7）鍵ラッピング対称鍵（Symmetric Key Wrapping Key）
8）乱数生成対称鍵（Symmetric RNG Key）
9）乱数生成プライベート鍵（Private RNG Key）
10）乱数生成公開鍵（Public RNG Key）
11）マスタ対称鍵（Symmetric Master Key）
12）鍵配送プライベート鍵（Private Key Transport Key）
13）鍵配送公開鍵（Public Key Transport Key）
14）鍵合意対称鍵（Symmetric Key Agreement Key）
15）鍵合意静的プライベート鍵（Private Static Key Agreement Key）
16）鍵合意静的公開鍵（Public Static Key Agreement Key）
17）鍵合意一時的プライベート鍵（Private Ephemeral Key Agreement Key）
18）鍵合意一時的公開鍵（Public Ephemeral Key Agreement Key）
19）認可対称鍵（Symmetric Authorization Key）
20）認可プライベート鍵（Private Authorization Key）
21）認可公開鍵（Public Authorization Key）

メタデータは、CKMS によって明示的に記録され管理されている特定の暗号鍵に関連付けられている情報として定義されるものであり、特性、制約、受け入れられるユーザ及び適用可能なパラメタを指定する。メタデータの各ユニットはメタデータ要素と呼ばれる。SP800-130 で取り上げる典型的なメタデータ要素は以下の通りである。

a）鍵ラベル（Key Label）
b）鍵識別子（Key Identifier）
c）所有者識別子（Owner Identifier）
d）鍵ライフサイクル状態（Key Lifecycle State）
e）鍵フォーマット指定子（Key Format Specifier）

f) 鍵生成に使用した製品（Product used to create the Key）
g) 鍵を使用する暗号アルゴリズム（Cryptographic Algorithm using the Key）
h) スキーム又は暗号利用モード（Scheme or Modes of Operation）
i) 鍵パラメタ（Parameters for the Key）
j) 鍵長（Length of the Key）
k) 鍵／アルゴリズム組のセキュリティ強度（Security Strength of the Key/Algorithm Pair）
l) 鍵タイプ（Key Type）
m) 鍵に対する適切なアプリケーション（Appropriate Applications for the Key）
n) 鍵セキュリティポリシー識別子（Key Security Policy Identifier）
o) 鍵アクセスコントロールリスト（Key Access Control List（ACL））
p) 鍵使用カウント（Key Usage Count）
q) 親鍵（Parent Key）
r) 鍵機微性（Key Sensitivity）
s) 鍵保護（Key Protections）
t) メタデータ保護（Metadata Protections）
u) 信頼関係保護（Trusted Association Protections）
v) 日時（Date Times）
w) 失効理由（Revocation Reason）

解説・考慮点

暗号鍵は、「特性」に応じて、対称鍵、公開鍵、プライベート鍵に大きく分類できる。

対称鍵は、暗号化に使用する暗号鍵と復号に使用する暗号鍵が同一であり、情報を所有又は共有するエンティティ（情報所有者や、情報送信者と情報受信者）のみが秘密裏に利用する。

公開鍵とプライベート鍵はペアで使用する。暗号の場合、公開鍵は暗号化に使用し、プライベート鍵は復号に使用する。署名と検証の場合、公開鍵は検証に使用し、プライベート鍵は署名に使用する。公開鍵は、不特定多数のエンティティに知られても良いが、プライベート鍵は復号や署名する単独のエンティティ（一人のユーザ）以外には知られることが無いようにする。

時間的な「オプション」として、暗号鍵を長期に渡り利用する場合は静的な暗号鍵と呼び、短期間しか使わない場合は一時的な暗号鍵と呼ぶ。例えば、接続ごとに(EC)DH[18]で鍵交換する場合、一回の接続（1 つのセッションやトランザクション）でしか使用しない鍵であるため、一時的な鍵である。

「用途」は、その暗号鍵の利用用途や目的を表している。この用途と、上記の特性とオプションに応じて、「暗号鍵管理システム設計指針（基本編）」に記載されている 21 種類の鍵タイプに分類でき、それぞれの鍵タイプの役割は、以下の通りである。

[18] 一時的(EC)DH は、一時的（Ephemeral）であることを示すため(EC)DHE と記述することもある。

1) 署名プライベート鍵：デジタル署名を生成するためのプライベート鍵である。署名は公開鍵アルゴリズムが使用され、この鍵と鍵ペアとなる 2)の署名検証公開鍵が存在する。適切に扱うことができれば、署名プライベート鍵は、メッセージやドキュメント、保存されたデータのソース認証と完全性認証を提供するほか、それらの否認防止をサポートするためにも、使用することができる。
2) 署名公開鍵：デジタル署名を検証するための鍵である。公開鍵アルゴリズムで使用される鍵ペアの公開鍵であり、メッセージ、ドキュメント又は保存されたデータのソース認証と完全性認証を提供するほか、それらの否認防止をサポートすることを目的としたデジタル署名を検証するために使用される。
3) 認証対称鍵：対称鍵アルゴリズムと共に使用され、通信セッション、メッセージ、文書又は保存されたデータの ID 認証と完全性認証を提供する。対称鍵アルゴリズムの認証暗号利用モードでは、一つの鍵が認証と暗号化の両方に使用されることになる。（SP 800-175B を参照）。
4) 認証プライベート鍵：エンティティの身元の保証（つまり、ID 認証）を提供するための鍵である。この認証は公開鍵アルゴリズムが使用され、この鍵と鍵ペアとなる 5)の認証公開鍵が存在する。認証された通信セッション又は何らかのアクションを実行するための認可を確立するときに使用される。
5) 認証公開鍵：エンティティの身元の保証（つまり、ID 認証）を提供するための公開鍵アルゴリズムで使用される鍵ペアの公開鍵である。認証された通信セッション又は何らかのアクションを実行するための認可を確立するときに使用される。
6) データ暗号化／復号対称鍵：対称鍵アルゴリズムを用いて、データの機密性保護（平文データの暗号化）をするための鍵である。同じ鍵が、機密性保護を解除（暗号文データの復号）するためにも使用される。対称鍵アルゴリズムの認証付き秘匿モードでは、一つの鍵がソース認証と暗号化の両方に使用される。
7) 鍵ラッピング対称鍵：対称鍵アルゴリズムを用いて、他の鍵を暗号化するための鍵である。鍵暗号化鍵と呼ばれることもある。鍵の暗号化に使用された鍵ラッピング鍵は、暗号化処理を元に戻す（つまり、暗号化された鍵を復号する）ためにも使用される。鍵を使用するアルゴリズムによっては、完全性保護を提供するために鍵を使用することもできる。
8) 乱数生成対称鍵：対称暗号方式を使用して乱数を生成するための鍵である。
9) 乱数生成プライベート鍵：公開鍵アルゴリズムを使って乱数を生成するためのプライベート鍵である。この鍵と鍵ペアとなる 10)の乱数生成公開鍵が存在する。
10) 乱数生成公開鍵：乱数を生成するための鍵である。9)の乱数生成プライベート鍵のペアとなる乱数生成公開鍵である。
11) マスタ対称鍵：対称暗号化方式を使用して他の対称鍵（データ暗号化鍵や鍵ラッピング鍵など）を導出するための鍵である。マスタ鍵は、鍵導出鍵とも呼ばれる。
12) 鍵配送プライベート鍵：公開鍵暗号アルゴリズムを使用してペアとなる公開鍵で暗号化

された鍵の復号に使用される鍵である。この鍵と鍵ペアとなる 13)の鍵配送公開鍵が存在する。鍵配送鍵は、通常、対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び場合によっては他の鍵材料（例えば、初期ベクトル）を確立するために使用される。

13) 鍵配送公開鍵：公開鍵アルゴリズムを使用して鍵を暗号化するために使用される鍵である。12)の鍵配送プライベート鍵のペアとなる公開鍵である。これらの鍵ペアは、対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び場合によっては他の鍵材料（例えば、初期ベクトル）を確立するために使用される。確立された鍵の暗号化形態は、後で鍵配送プライベート鍵を使用して復号するために保存できる。

14) 鍵合意対称鍵：対称鍵合意アルゴリズムを使用して、対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための鍵である。

15) 鍵合意静的プライベート鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための長期的なプライベート鍵である。この鍵と鍵ペアとなる 16)の鍵合意静的公開鍵が存在する。

16) 鍵合意静的公開鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための長期的な公開鍵である。15)の静的鍵合意プライベート鍵のペアとなる公開鍵である。

17) 鍵合意一時的プライベート鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための短期的なプライベート鍵である。対称鍵や鍵材料を確立するために一度だけ使用される。この鍵と鍵ペアとなる 18)の鍵合意一時的公開鍵が存在する。

18) 鍵合意一時的公開鍵：公開鍵アルゴリズムを使用して対称鍵（例えば、鍵ラッピング鍵、データ暗号化鍵、MAC 鍵など）及び他の鍵材料（例えば、初期ベクトル）を確立するための短期的な公開鍵である。対称鍵や鍵材料を確立するために一度だけ使用される。17)の鍵合意一時的プライベート鍵のペアとなる公開鍵である。

19) 認可対称鍵：対称暗号方式を使用してエンティティに権限を付与するための鍵である。認可鍵は、認可されたエンティティへのアクセス権限の監視と付与を担当する責任を負うエンティティ、及びリソースへのアクセスを求めるエンティティに知らされる。

20) 認可プライベート鍵：権限に対する所有者の権利を証明するために使用（例えば、デジタル署名を使用）される非対称鍵（公開鍵）の鍵ペアのプライベート鍵である。この鍵と鍵ペアとなる 21)の認可公開鍵が存在する。

21) 認可公開鍵：関連する認可プライベート鍵を知っているエンティティに対する権限を検証するために使用される非対称鍵（プライベート鍵）の鍵ペアの公開鍵である。20)の認可プライベート鍵のペアとなる公開鍵である。

メタデータは、暗号鍵を適切に管理するために、その暗号鍵に関連付けられている情報である。例えば、鍵の保護方法や有効期間などである。「暗号鍵管理システム設計指針（基本編）」では、メタデータの典型的な要素として a)～w)の 23 種類が記載されている。詳細は以下の通りである。

なお、全ての暗号鍵に対してメタデータが必要となるわけではなく、またメタデータが必要な場合であっても全ての適用可能なメタデータ要素と関連付ける必要は必ずしもないことに留意されたい。

a) 鍵ラベル（Key Label）：人間が読解可能なテキスト文字列で書かれた鍵の記述子のことである。例えば、“root CA Private Key 2022”や“Cryptrec Secret Key 2023”などである。
b) 鍵識別子（Key Identifier）：多数の鍵から特定の鍵を識別するための識別子のことである。CKMS は一般的にユニークな識別子を割り当てる。
c) 所有者識別子（Owner Identifier）：鍵を所有するエンティティの識別子のことである。
d) 鍵ライフサイクル状態（Key Lifecycle State）：「4.2 暗号鍵のライフサイクル」に記載した鍵の状態のことである。
e) 鍵フォーマット指定子（Key Format Specifier）：鍵のフォーマットの指定子のことである。例えば、RSA 公開鍵は法（modulus）と公開指数（public exponent）があり、その 2 つの値の格納順及び値の表記方法を指定する。Internet Engineering Task Force（IETF）は、(EC)DH、RSA、ECDSA などの鍵を格納するための方法を RFC 5208、RFC 5480、RFC 5958 で定義している。
f) 鍵生成に使用した製品（Product used to create the Key）：鍵生成に使用した製品情報のことである。
g) 鍵を使用する暗号アルゴリズム（Cryptographic Algorithm using the Key）：その鍵が使用する暗号アルゴリズムを指定する。例えば、ECDSA、RSA、AES、Camellia、HMAC-SHA256 などである。
h) スキーム又は暗号利用モード（Scheme or Modes of Operation）：その鍵が使用する暗号アルゴリズムを実行するための適用可能なスキーム又は暗号利用モードのことである。例えば、非対称アルゴリズムでは、有限体、binary field（標数 2 の体）又は楕円曲線（EC）上の離散対数問題の演算を指定する。対称アルゴリズムでは、ブロック暗号で使用される暗号利用モード（CBC、OFB、CCM、GCM 等）を指定する。詳細に関しては、［SP 800-38A］～［SP 800-38D］を参照されたい。
i) 鍵パラメタ（Parameters for the Key）：鍵のパラメタが存在する場合に指定する。例えば、ECDSA 鍵には（素数（p）、楕円曲線の係数（a, b）、生成元（G）、G の位数（n）、n のコファクタ（h））のドメインパラメタがある。
j) 鍵長（Length of the Key）：鍵の長さをビット（又はバイト）で指定する。例としては、RSA の法の 2048 ビットや楕円曲線暗号の鍵の 256 ビットである。
k) 鍵／アルゴリズム組のセキュリティ強度（Security Strength of the Key/Algorithm Pair）：「5.1.1 暗号アルゴリズムのセキュリティ強度」に記載したセキュリティ強度である。
l) 鍵タイプ（Key Type）：本節で説明している鍵タイプのことである。

m) 鍵に対する適切なアプリケーション（Appropriate Applications for the Key）：鍵を使用してよいアプリケーションのことである。例えば、TLS、SSH、デジタル署名である。

n) 鍵セキュリティポリシー識別子（Key Security Policy Identifier）：暗号鍵又は鍵タイプに適用できるセキュリティポリシーを特定する識別子のことである。このセキュリティポリシーは、生成から破壊までの鍵ライフサイクル全体にわたって、暗号鍵又は鍵タイプを保護するために使用するセキュリティコントロール一式である。

o) 鍵アクセスコントロールリスト（Key Access Control List（ACL））：暗号鍵及びメタデータの管理機能で制限された通りに、暗号鍵へのアクセスが可能なエンティティを特定するリストである（「7.1 鍵情報へのアクセスコントロール」を参照）。例えば、Microsoft Windows の ACL、Linux の ACL 、HSM を使った方法などがある。

p) 鍵使用カウント（Key Usage Count）：鍵が使用された回数のことである。

q) 親鍵（Parent Key）：メタデータに関連付けられた暗号鍵の導出元となった暗号鍵のことである。例えば、TLS 1.3 の通信に使われる暗号鍵である各種トラフィックシークレットは、マスタシークレットやハンドシェイクシークレットなどを親鍵として導出される。

r) 鍵機微性（Key Sensitivity）：暗号鍵の機微度又は重要度のことである。これは、リスクレベル（例えば、低、中、高）又は機密区分レベル（例えば、Confidential、Secret、Top Secret）に関係する。

s) 鍵保護（Key Protections）：暗号鍵に対する完全性、機密性及びソース認証（source authentication）の保護メカニズムのことである。公開鍵証明書は、CA のデジタル署名が完全性保護とソース認証の両方を提供する鍵保護の例である（［X.509］参照）。対称鍵及びそのハッシュ値を共に暗号化したものは機密性と完全性の保護の例である。なお、暗号鍵及びそのメタデータを外部エンティティから受信した場合は、それらが保護されているかどうかを事前に検証する必要があり、一般に、1 つの暗号機能（例：HMAC 又はデジタル署名）が完全性保護とソース認証の両方を提供するために使用される。

この保護メカニズムでは、いくつかの下位要素を持つことがある：

i. 完全性保護に使用されるメカニズム（例：ハッシュ値、MAC、又はデジタル署名）
ii. 機密性保護に使用されるメカニズム（例：鍵ラッピング、又は鍵配送）
iii. ソース認証に使用されるメカニズム（例：MAC、又はデジタル署名）
iv. 特定の非暗号学的な信頼プロセスによって実施される保護の表示

t) メタデータ保護（Metadata Protections）：関連付けられたメタデータの完全性、機密性及びソース認証を保護するために使用されるメカニズムのことである。一般には、鍵保護と同じメカニズムで保護するが、別のメカニズムで保護することもある。鍵保護と同様の下位要素を持つことがある。

u) 信頼関係保護（Trusted Association Protections）：暗号鍵とメタデータが正しく関連付けられていることを保証するために、その暗号鍵とメタデータとの信頼関係を保護するためのメカニズムのことである。上記の項目 s）で挙げられている保護によって、暗号鍵とメタデータがひとつの集約した項目として保護されている場合、信頼関係保護は暗黙的に提供されている。それ以外で信頼関係保護が必要な場合は、以下の項目が提供されるべきである：

i. 完全性保護に使用されるメカニズム（例：ハッシュ値、MAC、デジタル署名、又は信頼プロセス）
ii. ソース認証に使用されるメカニズム（例：暗号学的メカニズム又は非暗号学的な信頼プロセス）

v) 日時（Date Times）：暗号鍵の状態遷移のための日時のことである。例えば、鍵の使用開始日（活性化状態に遷移）や終了日（活性化状態から非活性化状態に遷移）、有効期限等に関するものがある。

w) 失効理由（Revocation Reason）：暗号鍵が失効した場合の失効理由のことである。例えば、暗号鍵の漏洩の疑い、暗号モジュール危殆化の疑い、鍵所有者の組織離脱、鍵の誤使用等がある。

4.2 鍵情報の選択

① CKMS が取り扱う全ての鍵タイプの利用用途及び生成手段、メタデータ、信頼関係、保護方針などの決定

項目	FR 番号	Framework Requirements の内容	SP800-130
D.01	FR6.1	CKMS 設計は、使用されているそれぞれの鍵タイプを明記及び定義しなければならない。	6.1 節
D.02	FR6.2	システムで使用されているそれぞれの鍵タイプに対して、CKMS 設計は、信頼関係のために選択される全てのメタデータ要素、メタデータ要素が作成され鍵との関連付けが満たされている状況、及び関連付けの手段（すなわち、暗号メカニズム又は信頼プロセス）を明記しなければならない。	6.2.1 節
D.03	FR6.13	それぞれの鍵タイプに対して、CKMS 設計は、暗号鍵及びメタデータ要素に関する以下の情報を明記しなければならない： a) 鍵タイプ b) 暗号鍵有効期間（cryptoperiod）（静的鍵（static key）に対して） c) 生成手段 i. 使用した乱数生成器（RNG） ii. 鍵生成の仕様（例えば、署名鍵については［FIPS 186］、Diffie-Hellman 鍵確立鍵（key establishment key）については［SP800-56A］） d) それぞれのメタデータ要素に対して、以下を含める i. メタデータのソース ii. メタデータの検証方法 e) 鍵確立（key establishment）の手段	6.2.2 節

		i. 鍵配送スキーム（使用されている場合） ii. 鍵合意スキーム（使用されている場合） iii. プロトコル名（名称があるプロトコルが使用されている場合） f) 暴露に対する保護（例えば、鍵の機密性、物理セキュリティ） g) 改ざんに対する保護（例えば、MAC 又はデジタル署名） h) 鍵を使用し得るアプリケーション（例えば、TLS、EFS、S/MIME、IPSec、PKINIT、SSH、等） i) 鍵の使用が許可されないアプリケーション j) 鍵保証（key assurances） i. 対称鍵保証（Symmetric key assurances）（例えば、フォーマットチェック） • 誰が保証を得るか • 保証が得られる状況 • どのように保証を得るか ii. 非対称鍵保証（Asymmetric key assurances）（例えば、所有と有効性の保証） • 誰が保証を得るか • 保証が得られる状況 • どのように保証を得るか iii. ドメインパラメタ有効性チェック • 誰が有効性チェックを実行するか • チェックが実行される状況 • どのようにドメインパラメタの有効性の保証を得るか	
D.04	FR6.14	CKMS 設計は、CKMS によって生成、保管、伝送、処理、及びその他管理される全ての鍵タイプ及びメタデータについて、全てのシンタックス、セマンティクス、及びフォーマットを明記しなければならない。	6.2.2 節

解説・考慮点

CKMS の設計にあたって、項目 D.01～D.04 は、鍵情報の選択にあたっての要求事項を明確化することを求めたものである。D.01 は利用する鍵タイプの一覧、D.02 はメタデータに関しての要求事項、D.03 は鍵情報の利用条件や取り扱い方法、D.04 は書式方法を対象にしている。

本節の要求事項で、「CKMS が取り扱う全ての鍵タイプの利用用途及び生成手段、メタデータ、信頼関係、保護方針などを決めなければならない」と求めているが、ここでの「全ての鍵」の意図は、CKMS 設計で「明示的に選択や管理する必要がある全ての鍵（タイプ）」に対してのことを指している。つまり、CKMS 設計で明示的に選択や管理しておらず、プロトコルや製品仕様により内部処理として自動的に生成・使用される暗号鍵は基本的には含まない。但し、このことは、

利用する製品やアプリケーション、システムが自動的に生成・使用される暗号鍵を「ブラックボックスとして使っている」という認識を持つことが重要である。したがって、このような処理を行う部分（多くは暗号モジュール）については信頼できる製品を使うことが望ましい。例えば、暗号モジュール認証を取得した製品などである。もし信頼性に確信が持てない暗号モジュールを使用している場合などは、可能であれば、内部の処理を調査し、暗号鍵の信頼性を確認することが望ましい。

具体的に、D.01 では、CKMS 設計者が明示的に選択して管理する全ての鍵タイプを洗い出し、どのような鍵タイプを利用しているのかを明確化することを求めている。その際、使用するソフトウェアの仕様書、設計書や規格などを参考にすべきである。市販品（COTS）デバイスやオープンソースソフトウェア（OSS）を使用している場合は、公開されている仕様書等を参照して記載すべき鍵タイプなどの情報を得ることができることもある。なお、システム内に他者が実装したサブモジュールや既存システム内に前任者等が設計したサブモジュールが含まれている場合には、CKMS 設計者が明示的に選択して管理する鍵情報として、それらのモジュールで使われる鍵情報も対象に含まれる場合がある。このようなケースでは、必要に応じて、例えば SBOM（Software Bill Of Materials : ソフトウェア部品表）などを利用して、実態調査を行うことも想定されたい。

暗号鍵の各種管理機能を実行するためには暗号鍵とメタデータが正しく関連付けられている必要がある。そのため、D.02 と D.03 では、D.01 で洗い出した個々の鍵タイプに対して、対象となる暗号鍵の完全性を確保するために必要となるメタデータの洗い出しと、その暗号鍵とメタデータが正しく関連付けられていることを保証するための方法を明確にすることが求められている。

D.02、D.03 における信頼関係や鍵の保護方法や利用手段の決定においては、システムで使用する OS やハードウェア環境により選択可能な方式が異なることに留意されたい。例えば暗号鍵の保管では、OS の標準的なファイル保護機能やプロセス保護機能（Microsoft Windows の ACL、Linux のパーミッションや ACL など）を利用する方法、強固な OS の機能（SELinux[19]など、Linux LSM[20]の保護）を利用する方法、TEE[21]に保管する方法、IC カードや TPM[22]に保管する方法、HSM[23]に保管する方法などがある。なお、これらの機能や方法が提供できるセキュリティ強度には違いがあるので、利用用途や想定される脅威等を踏まえて必要なセキュリティ強度を提供する機能や方法を選択することが重要である。

また、最近では、クラウド鍵管理 SaaS など外部の暗号鍵管理システムの保護方法などを利用することも考えられる。さらに、暗号鍵やメタデータの保護や信頼関係の保護では、前述する安全な暗号鍵の保管が可能な保護方法による信頼プロセスを利用する方法、又は MAC やデジタル

---

19 SELinux（Security-Enhanced Linux）: NSA により開発されたユーザ管理を詳細に制御できるようにした Linux のセキュリティ・アーキテクチャ

20 LSM（Linux Security Module）: Linux カーネルにセキュリティ機能を拡張するためのフレームワーク

21 TEE（Trusted Execution Environment）: チップ上に、通常の OS から独立した、信頼できる隔離実行可能な領域があり、その領域内で処理を完結させることができる環境のこと。その領域内に鍵の保存が可能である。Intel SGX、Arm TrustZone、AMD SEV、RISC-V Keystone などがある

22 TPM（Trusted Platform Module）: 暗号鍵を安全に格納できるセキュリティチップ。TCG（Trusted Computing Group）で規定された仕様には TPM1.2 及び TPM 2.0 が存在する

23 HSM（Hardware Security Module）: 耐タンパ性があり安全に暗号鍵管理や暗号処理をするモジュール

署名などの暗号メカニズムを使う方法を用いて実現する。

4.3 節との関係では、D.02、D.03 で信頼関係や鍵の保護方法や利用手段をどのように設定するかを高レベルの概要で整理し明らかにしておくことで、具体的な保護方法や利用手段について、2.3 節⑰及び 4.3 節に記載する内容との整合性を満たすことが重要である。ここでの「高レベルの概要」の意味は、4.3 節で決める事項を検討する際に本概要で定めたことと矛盾していないことが確認できる程度に具体化した情報、ということである。

加えて、D.03 で設定する情報や利用する各手段については、2 章で定める内容と矛盾なく、整合的であるようにする必要がある。例えば、鍵生成手段であれば 2.3 節⑧と、鍵確立手段であれば 2.5 節と、保管手段であれば 2.4 節①②と、鍵保証であれば 2.3 節⑩⑪とそれぞれ整合性が取れていることを確認することが重要である。

また、D.02 では、暗号鍵とメタデータの関連付けの手段として、両者に適切な暗号検証機能を適用し関連性が正しいことを検証する暗号メカニズム、又は物理的なセキュリティ手段により両者の関連性が正しいことを確認する信頼プロセスのいずれを利用しているか（または両方を利用しているか）を明らかにする。ここで、暗号メカニズムを選択した場合は、4.3 節の「暗号学的プロセスを利用するケース」に該当し、D.05～D.07 を記載することとなる。信頼プロセスを選択した場合は、「信頼プロセスを利用するケース」に該当し、D.08～D.10 を記載することになることに留意されたい。

D.04 は、主に自動処理する際に、全ての鍵タイプ及びメタデータが誤りなく適切に使われるようにするため、鍵タイプやメタデータの利用形態を統一化しておくための情報となる。これには、仕様書として明確化しておくほか、利用する標準規格や API などの情報を記載するなどのやり方がある。

これらの情報を記載しておくことにより、例えばある暗号鍵に対して危殆化が発生し、またその疑いが生じた場合においても、その危殆化の範囲をいち早く調査可能となることが期待される。また、監査等の効率も向上することが期待される。

《トイモデルと記載例》

本節のトイモデルでは、図 4-1 の通り、関係者のみがアクセス可能な施設内 Web サーバシステムとする。また、関係者のみが利用するシステムであることから、サーバ証明書を発行する認証局としては施設内にプライベート CA を独自に構築する形をとっている。ただし、トイモデルとしての説明を簡単にするため、ここでは CKMS の対象範囲を Web サーバに絞って定めるものとし、プライベート CA における暗号鍵管理については別途検討するものとする[24]。

[24] プライベート CA での暗号鍵管理においては、設置場所や運用環境、目的などに依存して、要求されるセキュリティレベルは大きく異なる。認証局のセキュリティとして最も高いレベルは、多くの民間認証局（パブリック CA）などが使っている WebTrust CA の基準（WebTrust Principles and criteria）に準拠する水準であるが、プライベート CA ではそこまでの水準を求められるケースはかなり少ない。なお、WebTrust CA の基準に近いセキュリティを確保することが求められるようなケースでは、CA 運用上どのような点に注意する必要があるかを検討する際に WebTrust のガイダンスが参考になる。
https://www.cpacanada.ca/en/business-and-accounting-resources/audit-and-assurance/overview-of-webtrust-services/principles-and-criteria

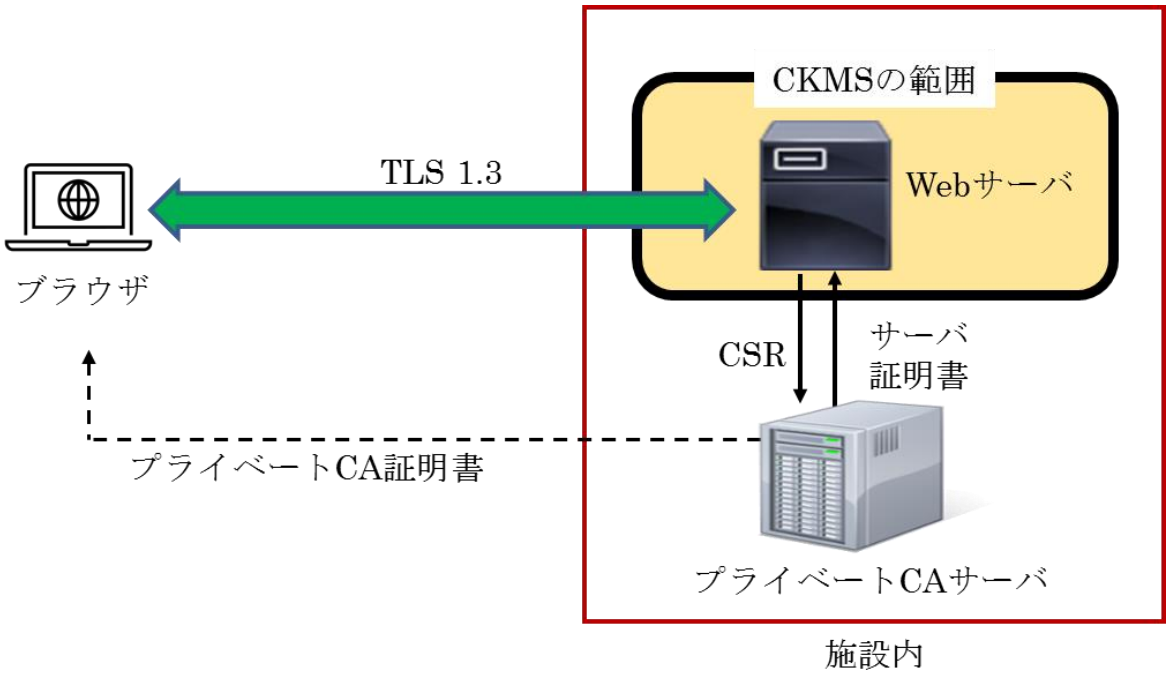


図 4-1　施設内 Web サーバシステム

Web サーバは、OS に Linux、Web サーバの機能を実現するソフトウェアに Apache HTTP Server を使用する。セキュリティプロトコルとして TLS 1.3 だけを許可し、TLS 1.3 を実現するソフトウェアには OpenSSL を使用する。

TLS における暗号鍵を適切に設定するために「TLS 暗号設定ガイドライン」などを参照し、証明書の署名方式は ECDSA NIST P-256 を使用し、鍵交換（鍵合意）の ECDH は X25519、X448、P-256、P-386、P-521 を使用する。なお、この設定は「TLS 暗号設定サーバ設定編」の Apache に関係する部分も参考にしている。

D.01 に明記する鍵タイプは、解説・考慮点に記載したように、CKMS 設計者が明示的に選択して管理する鍵である。本モデルでは、Web サイト正当性のための Web サーバの署名用のプライベート鍵と公開鍵が記載対象となる。また、TLS 1.3 の鍵交換（鍵合意）の ECDH 関係の鍵も記載対象となる。なお、OpenSSL 内部で生成する TLS セッションで利用される鍵は管理対象外となる。また、Web サーバの署名用のプライベート鍵と公開鍵は、PKCS#8（RFC 5958）形式のパスワード付き暗号ファイルで保存する。この暗号化には AES128 ビットの CBC モードを利用する。

ここで使用する証明書はサイトの正当性のためなので、証明書の key usage は電子署名（Digital Signature）を含まなければならない。なお、証明書はここで指定した用途以外では使用できないことに留意されたい。

以上のトイモデルにおける記載例は、以下の「施設内 Web サーバシステムにおける記載例」のようになる。

施設内 Web サーバシステムにおける記載例

D.01	[Web サーバが利用する鍵タイプ] ● 署名アルゴリズム：ECDSA（P-256） ● 署名プライベート鍵（Private Signature Key） ● 署名公開鍵（Public Signature Key） [鍵交換（鍵合意）で利用される鍵タイプ] ● 鍵交換（鍵合意）アルゴリズム：ECDH（X25519, X448, P-256, P-386, P-521） ● 鍵合意一時的プライベート鍵（Private Ephemeral Key Agreement Key） ● 鍵合意一時的公開鍵（Public Ephemeral Key Agreement Key）
D.02	● 署名プライベート鍵は、PKCS#8（RFC 5958）形式のパスワード付き暗号ファイルで保存する。ファイル暗号化には AES128-CBC を利用する。このファイルは Linux の root 権限で保存し、root 権限の無いユーザは Linux の ACL でアクセスすることができないようにしている ● 署名公開鍵の完全性保護のために、証明書に記載されている署名公開鍵とメタデータ（subjectAltName, subject key id など）は暗号メカニズム（デジタル署名）により関連付けられる
D.03	[署名プライベート鍵と署名公開鍵] a) 鍵タイプ ● 署名プライベート鍵（Private Signature Key） ● 署名公開鍵（Public Signature Key） b) 暗号鍵有効期間 ● 証明書の有効期間（署名プライベート鍵、署名公開鍵の有効期間） ● 日本時間 2023 年 5 月 8 日 00:00:00 ～ 日本時間 2024 年 6 月 7 日 23:59:59 c) 生成手段 i) 使用した乱数生成器（RNG）：Linux の乱数生成の疑似デバイスである /dev/random を使用する ii) 鍵生成の仕様：FIPS186-4 d) それぞれのメタデータ要素に対して、以下を含める i) メタデータのソース メタデータ要素（common name, 組織名、有効期間、subjectAltName, subject key id など）のソースは、証明書を作成したときの Web サーバの FQDN、IP アドレス、使用する鍵やアルゴリズムなどの各種設定情報により決定する ii) メタデータの検証方法 証明書に記述されているメタデータの内容検証のために、プライベート CA から始まる証明書パスの有効性検証、及び FQDN と IP アドレスの一致確認を実施する e) 鍵確立（key establishment）の手段：対象外 f) 暴露に対する保護（例えば、鍵の機密性、物理セキュリティ） 署名プライベート鍵は PKCS#8（RFC 5958）形式のパスワード付き暗号化状態で

	保存している。このファイルは Linux の root を所有者にして保存し、ルート以外のユーザは Linux の ACL でアクセスすることができないようにしている g) 改ざんに対する保護（例えば、MAC 又はデジタル署名） 証明書はプライベート CA によるデジタル署名により保護する h) 鍵を使用し得るアプリケーション Web Server（Apache HTTP Server）、TLS 1.3 i) 鍵の使用が許可されないアプリケーション h)で指定したアプリケーション以外は使用を許可しない j) 鍵保証（key assurances） i) 対称鍵保証（Symmetric key assurances） 該当無し ii) 非対称鍵保証（Asymmetric key assurances） 証明書発行者（Issuer）の署名により署名プライベート鍵を保証する iii) ドメインパラメタ有効性チェック 証明書の ECDSA 署名で使用している楕円曲線 P-256 はパラメタを OID で示している。ドメインパラメタの値自体を保管、伝送しないので、有効性は保証される [鍵合意一時的プライベート鍵と公開鍵] a) 鍵タイプ ● 鍵合意一時的プライベート鍵（Private Ephemeral Key Agreement Key） ● 鍵合意一時的公開鍵（Public Ephemeral Key Agreement Key） b) 暗号鍵有効期間 一時的 c) 生成手段 i) 使用した乱数生成器（RNG）：Linux の乱数生成の疑似デバイスである /dev/random を使用 ii) 鍵生成の仕様：SP800-56A Rev.3 d) それぞれのメタデータ要素に対して、以下を含める i) メタデータのソース TLS 1.3 のプロトコル ii) メタデータの検証方法 TLS 1.3 のプロトコルで検証 e) 鍵確立（key establishment）の手段 i) 鍵配送スキーム：使用していない ii) 鍵合意スキーム：ECDH（X25519, X448, P-256, P-386, P-521） iii) プロトコル名：TLS 1.3 f) 暴露に対する保護（例えば、鍵の機密性、物理セキュリティ） 鍵は、OpenSSL が動作するプロセスのメモリ上で一時的に使用され即破棄される g) 改ざんに対する保護（例えば、MAC 又はデジタル署名）

	鍵合意一時的プライベート鍵は OpenSSL が動作するプロセスのメモリ上だけで使用されることにより保護される。鍵合意一時的公開鍵も受信後はメモリ上で処理されることにより保護される h) 鍵を使用し得るアプリケーション Web Server（Apache HTTP Server)、TLS 1.3 i) 鍵の使用が許可されないアプリケーション h)で指定したアプリケーション以外は使用を許可しない j) 鍵保証（key assurances） i) 対称鍵保証（Symmetric key assurances） 該当無し ii) 非対称鍵保証（Asymmetric key assurances） ECDH の場合、当該楕円曲線上の点であり、特異な点でないことを確認する iii) ドメインパラメタ有効性チェック 鍵合意は、TLS 1.3 で規定されているドメインパラメタを使用している。規定された番号で合意し、ドメインパラメタの値自体を伝送しないので、有効性は保証される
D.04	● 署名のプライベート鍵と公開鍵の保管方法は PKCS#8 形式のパスワード付き暗号化状態で保存する ● 証明書は X.509 を使用する。伝送は、TLS 1.3 を使用する

4.3 鍵情報の保護方針

① メタデータ要素内に含まれている情報の保護方法の決定

● 暗号学的プロセスを利用する場合

項目	FR 番号	Framework Requirements の内容	SP800-130
D.05	FR6.3	メタデータ要素の鍵保護（Key Protections）で使用されるそれぞれの暗号メカニズムに対して、CKMS 設計は、以下を明記しなければならない： i. 暗号アルゴリズム ii. 鍵パラメタ iii. 鍵識別子 iv. 保護値（protection value）：この要素は、完全性保護、機密性保護、又はソース認証（source authentication）の保護値（protection value）を含む。例えば、適切に実装された MAC 又はデジタル署名技術は、完全性保護やソース認証（source authentication）を提供し得る。 v. 保護が適用された時期	6.2.1 節

		vi. 保護が検証された時期	
D.06	FR6.5	メタデータ要素のメタデータ保護（Metadata Protections）で使用されるそれぞれの暗号メカニズムに対して、CKMS 設計は、以下を明記しなければならない： i. 暗号アルゴリズム ii. 鍵パラメタ iii. 鍵識別子 iv. 保護値（protection value）（例：MAC、デジタル署名） v. 保護が適用された時期 vi. 保護が検証された時期 一般に、特に鍵とメタデータがひとまとめにされる場合、鍵とメタデータに対して同じメカニズムが使用される。	6.2.1 節
D.07	FR6.7	メタデータ要素の信頼関係保護で使用されるそれぞれの暗号メカニズムに対して、CKMS 設計は、以下を明記しなければならない： i. 暗号アルゴリズム ii. 鍵パラメタ iii. 鍵識別子 iv. 保護値（protection value）（例：MAC、デジタル署名） v. 保護が適用された時期 vi. 保護が検証された時期	6.2.1 節

- 信頼プロセスを利用する場合

項目	FR 番号	Framework Requirements の内容	SP800-130
D.08	FR6.4	メタデータ要素の鍵保護（Key Protections）で使用される暗号学的ではないそれぞれの信頼プロセスに対して、CKMS 設計は、以下を明記しなければならない： i. 他のプロセスと区別するために使用されるプロセス識別子 ii. プロセスの説明又はプロセスの説明へのポインタ	6.2.1 節
D.09	FR6.6	メタデータ要素のメタデータ保護（Metadata Protections）で使用される暗号学的ではないそれぞれの信頼プロセスに対して、CKMS 設計は、以下を明記しなければならない： i. このプロセスを他のプロセスから区別するために使用される識別子 ii. プロセスの説明又はプロセスの説明へのポインタ	6.2.1 節
D.10	FR6.8	メタデータ要素の信頼関係保護で使用される暗号学的ではないそれぞれの信頼プロセスに対して、CKMS 設計は、以下を明記しなければならない：	6.2.1 節

		i. このプロセスを他のプロセスから区別するために使用される識別子 ii. プロセスの説明又はプロセスの説明へのポインタ	

解説・考慮点

メタデータ要素内の暗号鍵、メタデータ及びそれらの信頼関係に関して、暗号学的プロセス又は信頼プロセスのいずれかにより保護しなければならない。
CKMS の設計にあたって、項目 D.05～D.07 は、暗号学的プロセスを利用する場合の保護方法に関する要求事項を明確化することを求めたものである。D.05 は暗号鍵の保護、D.06 はメタデータの保護、D.07 は信頼関係の保護を対象にしている。
項目 D.08～D.10 は、信頼プロセスを利用する場合の保護方法に関する要求事項を明確化することを求めたものである。D.05～D.07 と同様、D.08 は暗号鍵の保護、D.09 はメタデータの保護、D.10 は信頼関係の保護を対象にしている。
なお、D.05～D.07、D.08～D.10 のいずれかが対象である。

CKMS 設計者が管理する必要がある全ての暗号鍵とメタデータは、鍵タイプや使用用途により適切な保護が必要であり、例えば、プライベート鍵は復号や署名するエンティティ以外には知られることが無いように保管することが必要である。また、鍵タイプによってはメタデータとの組み合わせによりその暗号鍵の完全性を確保する場合もあるため、とりわけそのような鍵タイプの暗号鍵ではメタデータの管理も安全な鍵情報の管理を実施するために重要となる。そのため、D.02 で暗号鍵とメタデータの関連付けの手段を利用するとしたメタデータについては、どのような関連付けの手段を利用し、両者の関連性が正しいことを保証するかについての具体的な方法を示すことが求められる。

本節では、暗号鍵やメタデータ、信頼関係の保護方針を明確にすることで、鍵情報が安全に管理されていることを確認することを意図している。

具体的には、保護手段として暗号メカニズムを使用している場合は、そこで利用している具体的な暗号アルゴリズムや鍵パラメタなど、D.05～D.07 の項目を明確化する必要である。D.05 は暗号鍵に対する、D.06 は関連付けられたメタデータに対する、完全性、機密性及びソース認証（source authentication）の保護が対象であり、D.07 は、暗号鍵とメタデータが正しく関連付けられていることを保証するために、その暗号鍵とメタデータとの信頼関係の保護が対象である。なお、これらは同じメカニズムで保護される場合もあれば、別のメカニズムで保護することもある。さらには、暗号鍵とメタデータが同時に関連付けて保護され、信頼関係保護が暗黙的に提供されている場合もある。

保護手段として信頼プロセスを使用している場合は、そこで利用している具体的なプロセスについて D.08～D.10 の項目を明確化することが必要である。

なお、以上の内容は、D.02、D.03 での記載内容と整合していなければならない。

《トイモデルと記載例》

本節のトイモデルは 4.2 節と同じである。このトイモデルでは、D.02 において、「信頼プロセスはプライベート鍵の保護のために root 権限でプライベート鍵ファイルを保存し、OS のユーザ管理機能で root 以外のユーザはファイルにアクセスできなくし、暗号学的プロセスはデジタル署名を利用して公開鍵やメタデータの完全性を保護する」こととしている。

その方針に従い、本トイモデルが利用している、具体的な暗号メカニズムを使用した保護手段についての情報を D.05～D.07 に、具体的な信頼プロセスを使用した保護手段についての情報を D.08～D.010 にそれぞれ記載する。

以上のトイモデルにおける記載例は、以下の「施設内 Web サーバシステムにおける記載例」のようになる。

施設内 Web サーバシステムにおける記載例

D.05	署名公開鍵の保護 i. 暗号アルゴリズム ECDSA ii. 鍵パラメタ P-256 iii. 鍵識別子 公開鍵のハッシュ値 iv. 保護値（protection value） デジタル署名 v. 保護が適用された時期 日本時間 2023 年 5 月 8 日 00:00:00 vi. 保護が検証された時期 日本時間 2023 年 5 月 8 日 00:00:00
D.06	署名の鍵ペアのメタデータの保護 i. 暗号アルゴリズム ECDSA ii. 鍵パラメタ P-256 iii. 鍵識別子 公開鍵のハッシュ値 iv. 保護値（protection value） デジタル署名 v. 保護が適用された時期 日本時間 2023 年 5 月 8 日 00:00:00 vi. 保護が検証された時期 日本時間 2023 年 5 月 8 日 00:00:00
D.07	署名の鍵ペアとメタデータの関係性の保護

	i. 暗号アルゴリズム ECDSA ii. 鍵パラメタ P-256 iii. 鍵識別子 公開鍵のハッシュ値 iv. 保護値（protection value） デジタル署名 v. 保護が適用された時期 日本時間 2023 年 5 月 8 日 00:00:00 vi. 保護が検証された時期 日本時間 2023 年 5 月 8 日 00:00:00

D.08	Web サーバでの署名プライベート鍵のファイルは、root 権限でのみアクセス可能。このアクセス権限は Linux のパーミッションを利用して実現している
D.09	Web サーバでの署名処理の入力として利用されるメタデータの管理に関するファイルは、root 又は Web サーバ管理ユーザ権限でのみアクセス可能。このアクセス権限は Linux のパーミッションを利用して実現している
D.10	D.09 での保護方法により、メタデータ要素の信頼関係も保護している

発行日　2023 年 5 月 9 日　　第 1.0 版発行
　　2025 年 4 月 25 日　　第 1.1 版発行

発行者

・〒113-6591
東京都文京区本駒込二丁目 28 番 8 号
独立行政法人 情報処理推進機構
（セキュリティセンター　技術評価部　暗号グループ）
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN
2-28-8 HONKOMAGOME, BUNKYO-KU
TOKYO, 113-6591 JAPAN

・〒184-8795
東京都小金井市貫井北町四丁目 2 番 1 号
国立研究開発法人 情報通信研究機構
（サイバーセキュリティ研究所　セキュリティ基盤研究室）
NATIONAL INSTITUTE OF
INFORMATION AND COMMUNICATIONS TECHNOLOGY
4-2-1 NUKUI-KITAMACHI, KOGANEI
TOKYO, 184-8795 JAPAN