CRYPTREC GL-3003-1.0.1

暗号鍵設定ガイダンス

2022 年 3 月

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
2022.07.01	初版発行
2023.03.30	発行月の誤り訂正 CRYPTREC 暗号リストのリンク先を追記

1. はじめに

1.1 本書の内容及び位置付け

情報を安全に取り扱うためには、通信情報や保管情報の暗号化や署名などに使う暗号技術のみに注意を払うだけでは不十分であり、その暗号技術に用いられる暗号鍵に対して適切に鍵長を設定し、さらに適切に鍵管理を行って安全に運用していくことが必要である。

図 1は、安全な暗号技術の導入にあたって考慮すべき項目と暗号技術の安全な運用にあたって考慮すべき項目の代表的なものを示しており、線でそれらの項目間での関連性を示している。本書では、このうち、緑色でハッチングされている項目を取り上げている。

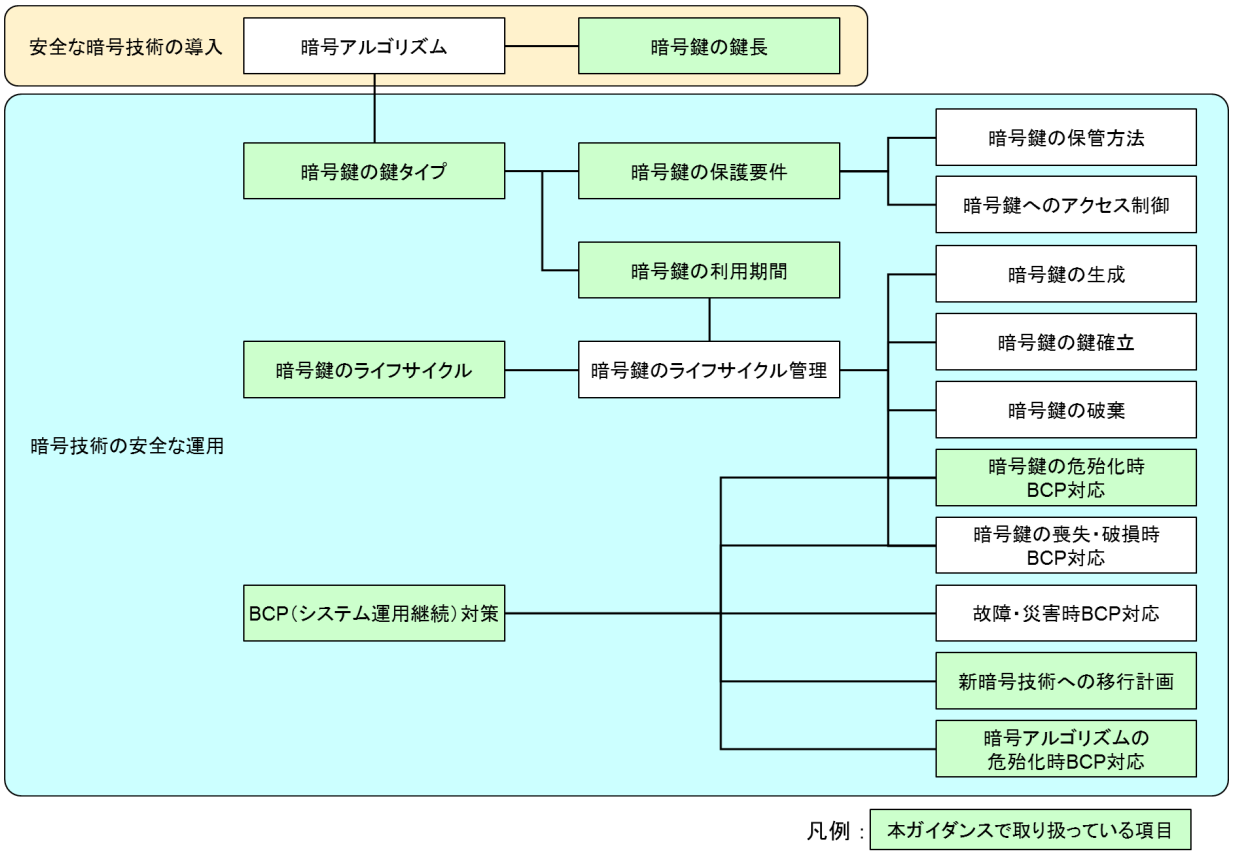

図 1　暗号技術の導入・運用にあたって考慮すべき代表的な項目

本書では、まず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する。

本書で示すセキュリティ強度は暗号技術のセキュリティ（暗号学的安全性）[1]を判断する上での

1　一般には「(暗号の）安全性」と表現されることが多いが、「安全性」には「物理的な安全性」や「人命などに対する安全性」といった意味で使われることもある。そのため、本書では、「(暗号の）安全性」のことを「セキュリティ」又は「暗号学的安全性」と表記する。

目安となるものであり、利用する鍵長によってセキュリティ強度と処理効率などが変わることに留意する必要がある。アルゴリズムの中には（特に RSA などの公開鍵暗号では）必要以上に長い鍵長を使用すると処理効率などに悪影響が出る場合がある一方、短すぎる鍵長を使用すると十分なセキュリティ強度を提供しないので、システムやアプリケーションの設計・開発にあたっては、適切なセキュリティ強度を満たすように鍵長を定めることが重要である。なお、実際の設計・開発にあたっては、鍵長以外の対策を適切に併用することによってシステム全体としてのセキュリティ確保を図るという方針を採用するも可能である。

本書を参考に、実際の利用用途や利用期間、環境、コスト、その他様々な制約条件を踏まえて、必要なセキュリティ強度を満たすように鍵長を**設定すべきである**。

次に、暗号技術の安全な運用の観点から、適切に暗号鍵の管理を行うために必要となる項目についての技術的概要を示す。なお、本書では具体的な対策方法や実現方法などについて説明しないので、より詳細な情報が必要であれば、NIST SP800-57 パート 1 改訂 5 版 [2]などを参考にされたい。

本書は 7 節で構成されており、節立ては以下の通りである。

1 節では、イントロダクションとして、本書の位置づけや想定読者を示す。
2 節では、本書を理解する上での技術的な基礎知識を説明する。また、暗号技術ごとの推定セキュリティ強度をまとめる。
3 節では、鍵長選択の考え方を記載する。
4 節では、鍵を安全に運用するために重要な、鍵の生成から破棄までのライフサイクルについて説明する。
5 節では、鍵の利用期間について考え方を提示する。
6 節では、鍵の保護について考慮すべきポイントを提示する。
7 節では、運用中における鍵長移行に関する検討の必要性を示し、その際の論点等を記載する。

【重要な注意①】

「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 [3]」に従ったセキュリティ強度（鍵長）を設定しない場合には、電子政府推奨暗号リスト [4]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すること。

【重要な注意②】

大規模な量子コンピュータが利用可能になった場合、Shor のアルゴリズムにより多項式時間で

---

2 NIST SP800-57 Part 1 Revision5 の日本語訳、https://www.ipa.go.jp/files/000090943.pdf
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、https://www.cryptrec.go.jp/list/cryptrec-ls-0003-2022.pdf
4 総務省・経済産業省、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)、https://www.cryptrec.go.jp/list.html

素因数分解問題や（楕円）離散対数問題が解けることが知られており、とりわけ CRYPTREC 暗号リストの公開鍵暗号（守秘、署名、鍵共有）に掲載されている全てのアルゴリズムにとって理論的には大きな脅威になっている [5]。

しかし、2021 年 3 月時点の CRYPTREC 調査 [6]では、35（＝5×7）の素因数分解が成功しなかったという研究発表などを踏まえ、「現状の量子コンピュータでは暗号で用いるほど大きなパラメータの合成数を素因数分解することは困難であり、暗号で用いるパラメータの問題を解くためには量子ビット数やゲート計算のエラー率など量子コンピュータの性能の大幅な向上が必要であると考える。」と結論付けている。このことは、現時点で実現されている量子コンピュータと実際の暗号解読を行うのに必要とされる量子コンピュータの性能に関しては依然として大きな乖離があることを意味している。加えて、量子コンピュータの性能を測る上での指標（量子ビット数、量子誤りの大きさ、演算可能回数など）や量子コンピュータの開発状況を考慮すると、本書の発行時点（2022 年 6 月）において量子コンピュータによる公開鍵暗号の危殆化時期を予測することは困難である。

したがって、**本書では量子コンピュータによる暗号技術の危殆化は将来的なリスク要因に位置づけている**。7.5 節も参照されたい。

1.2 本書が対象とする読者

本書は、暗号技術を組込んだシステム又はアプリケーションの設計・開発・運用・提供にあたって、安全な暗号技術の選定、及び暗号技術の安全な運用方針・対策の作成や決定などに携わる管理者、設計者、開発者などを主な想定読者とする。

---

5 共通鍵暗号、暗号利用モード、メッセージ認証コードに対しては、おおむね鍵長の半分程度のセキュリティ強度に低下するが、公開鍵暗号ほど大きな影響は受けないと評価されている。つまり、鍵長を 256 ビットにするなどの対策で対処可能である。詳細については、CRYPTREC Report 2019「量子コンピュータが共通鍵暗号の安全性に及ぼす影響の調査及び評価」を参照されたい。
https://www.cryptrec.go.jp/report/cryptrec-rp-2000-2019r1.pdf

6 CRYPTREC Report 2020「Shor の量子アルゴリズムによる現代暗号への脅威に関する調査」を参照されたい。https://www.cryptrec.go.jp/report/cryptrec-rp-2000-2020.pdf

2. 技術的な基礎知識

暗号鍵を安全に設定し、運用していくために考慮すべき項目として以下のものがある。

- 暗号鍵の鍵長（2.2 節、3 節）
  システムやアプリケーションなどのセキュリティ（暗号学的安全性）に直接影響する項目である。どのような暗号処理に使う鍵なのか（鍵タイプ）、どのぐらいの期間利用するものなのかに依存して適切な鍵長を選択する必要がある。
- 暗号鍵の鍵タイプ（2.1 節）
  どのような暗号処理に使う暗号鍵なのかによって決まる項目である。鍵タイプの種類によって、暗号鍵に求められるセキュリティ要件は異なる（5 節、6 節参照）。
- 暗号鍵のライフサイクル（4 節）
  個々の暗号鍵の生成から破棄までの鍵状態とその間の遷移を示す項目である。暗号鍵の取扱いで重要なのは、利用する様々な鍵に対して、この鍵のライフサイクルを正しく運用管理することである。とりわけ、利用期間が経過した暗号鍵を利用停止・破棄することや、暗号鍵の危殆化が起きた又は疑われる場合の当該鍵の利用を制限・破棄することなどを適切に行うことによって、暗号鍵そのものの安全性を確保することが必要である。

2.1 暗号処理及び鍵タイプの種類

本書で取り上げる暗号処理及び鍵タイプは、表 1 及び

表 2 の通りである。

表 1　暗号処理の種類及び鍵タイプの種類

暗号処理		概要	利用する鍵タイプ
暗号化	通信時	2 つ又はそれ以上のエンティティ（ユーザやデバイス等）間の通信路上での盗聴を防止することを目的とした処理のこと。「暗号通信」ともいう。 送信者がデータの暗号化を行うタイミングと受信者が暗号化された通信データを復号するタイミングは時間的にそれほど離れていないことを前提とする。つまり、暗号化された通信データがそのまま長期間保存されることは想定しない。	データ暗号化対称鍵
	保管時	データベースやストレージデバイスなどに保管されるデータの機密性保護を目的とした処理のこと。 長期にわたって安全な保管ができるようにすることが期待され、データの暗号化を実施するタイミングと、復	

		号してデータを取り出すタイミングが大きく異なることが想定される。	
	鍵共有	共通鍵暗号を用いた暗号通信に先立ち、2 つ又はそれ以上のエンティティ間で、盗聴されずにセッション鍵の共有・確立・合意を行い、当該エンティティ間でセッション鍵を安全に共有することを目的とした処理のこと。	鍵共有対称鍵 鍵共有公開鍵 鍵共有プライベート鍵
署名		対象データの完全性及び署名者の検証を行い、当該データの完全性を確保することを目的とした処理のこと。当該データの否認防止の確認にも寄与する。 有効な（失効していない）署名検証用の公開鍵証明書の有効期間（*NotBefore* から *NotAfter* の期間）内では、当該データの完全性及び署名者の正当性が確保されることが期待される。	署名プライベート鍵 署名検証公開鍵
メッセージ認証		通信データや保管データの完全性検証を行い、当該データが変更されていないことを確認することを目的とした処理のこと。	認証対称鍵
エンティティ認証		正規のエンティティであることを確認することを目的とした処理のこと。	認証対称鍵 認証プライベート鍵 認証公開鍵

表 2　鍵タイプの種類

鍵タイプ	概要
データ暗号化対称鍵	共通鍵暗号を用いて、データの機密性保護に適用（平文データを暗号化）するために使用される。また、同じ鍵が、機密性保護を解除（暗号文データを復号）するためにも使用される。 この鍵は、暗号化したデータの機密性保護が必要な期間、秘密に**保持されなければならない**。
鍵共有対称鍵	共通鍵暗号を用いて、セッション鍵等の鍵情報を暗号化するために使用される。また、同じ鍵が暗号化された鍵情報を復号するためにも使用される。 この鍵は、暗号化した鍵情報の機密性保護が必要な期間、秘密に**保持されなければならない**。 鍵の名称については、鍵暗号化鍵、鍵ラッピング鍵、鍵合意対称鍵、鍵交換対称鍵と呼ばれることがある。
鍵共有公開鍵 鍵共有プライベート鍵	公開鍵暗号を用いて、セッション鍵等の鍵情報を交換するために使用される。鍵情報は、鍵共有プライベート鍵に対応する鍵共有公開鍵で暗号化され、当該鍵共有プライベート鍵で復号される。 鍵共有プライベート鍵は、鍵情報の交換が有効な期間中、秘密に**保持**

	されなければならない。 同じデータを暗号化すると同じ暗号文になるやり方で鍵共有を行う静的（長期的）な方式と、同じデータを暗号化しても異なる暗号文になるやり方で鍵共有を行う一時的（短期的）な方式とがある。 また、鍵共有の形式により、鍵合意、鍵配送、鍵交換などと呼ばれることがある。鍵の名称についても、鍵交換プライベート鍵、鍵交換公開鍵と呼ばれることもある。
署名プライベート鍵 署名検証公開鍵	公開鍵暗号を用いて、署名生成及び署名検証を行うために使用される。署名は、署名プライベート鍵を使って生成され、それに対応する署名検証公開鍵を使って検証される。 署名検証公開鍵は公開鍵証明書により、対応する署名プライベート鍵との関係が保証され、当該公開鍵証明書の有効期間中は署名検証に用いることができる。 署名プライベート鍵は、署名生成が許可されている期間中、秘密に**保持されなければならず**、同期間終了後は遅滞なく**破棄されなければならない**。もし、その期間中に漏えい・紛失等をした場合には、直ちに対応する署名検証公開鍵の公開鍵証明書を**失効させるべきである**。
認証対称鍵	共通鍵暗号（多くの場合、ブロック暗号）を用いて、データの完全性認証、又は ID 認証／エンティティ認証を行うために使用される。認証するデータに対する認証コードの生成及び検証に同じ鍵が用いられる。 この鍵は、認証が必要な期間、秘密に**保持されなければならない**。
認証プライベート鍵 認証公開鍵	公開鍵暗号を用いて、ID 認証／エンティティ認証を行うために使用される。一般にチャレンジ&レスポンス方式での認証が行われ、レスポンスを生成する際に認証プライベート鍵を利用し、当該レスポンスの正当性を確認する際に対応する認証公開鍵が利用される。 認証公開鍵は公開鍵証明書により、対応する認証プライベート鍵との関係が保証され、当該公開鍵証明書の有効期間中は ID 認証／エンティティ認証に用いることができる。 認証プライベート鍵は、認証が必要な期間、秘密に**保持されなければならない**。もし、その期間中に漏えい・紛失等をした場合には、直ちに対応する認証公開鍵の公開鍵証明書を**失効させるべきである**。

2.2 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

技術分類が異なる暗号技術のアルゴリズムについて、同じ程度のセキュリティ（暗号学的安全性）を有するかどうかを判断する目安として、“ビットセキュリティ”（等価安全性ということもある）という指標がある。具体的には、評価対象とするアルゴリズムに対して最も効果的な攻撃

手法を用いたときに、どの程度の計算量があれば解読できるか（解読計算量 [7]）に関連付けられた値で、鍵長とは別に求められる。表記上、解読計算量が 2^x である場合に“x ビットセキュリティ”という。

表 3～表 5 に、CRYPTREC 暗号リストに掲載されている暗号技術について、一般的に使用されているビットセキュリティ（112 ビット、128 ビット、192 ビット及び 256 ビット）を実現していると評価（推定）されている鍵長をアルゴリズムごとに示す。

ビットセキュリティによる評価では、技術分類に関わらず、どのアルゴリズムであっても、解読計算量が大きければセキュリティ（暗号学的安全性）が高く、逆に小さければセキュリティ（暗号学的安全性）が低い。また、解読計算量が実現可能と考えられる計算機能力を大幅に上回っていれば、少なくとも現在知られているような攻撃手法ではそのアルゴリズムを破ることは現実的に不可能であると期待される。

ただし、これらのビットセキュリティの推定値は、本書の発行時点（2022 年 6 月）で知られている最良の攻撃方法を用いた際の研究結果に基づいている。そのため、数体篩法、指数計算法、ρ 法といった素因数分解問題や（楕円）離散対数問題の解法アルゴリズムの進展はもとより、全く新しい解法アルゴリズムの登場や大規模な量子コンピュータの実用化などによって、ビットセキュリティの推定値が今後見直される可能性があることに留意されたい。推定値の妥当性を確認する観点から、本書は少なくとも 5 年ごとに（必要があれば適宜）記載内容の再レビューを実施するものとし、必要に応じて適切な修正を加えることを計画している。

【重要な注意】

1.1 節に記載の通り、量子コンピュータによる暗号技術の危殆化は将来的なリスク要因に位置づけているので、**推定セキュリティ強度の評価に量子コンピュータの影響は考慮していない**。また、**鍵長の選択要件においてもその影響を考慮しないものとする**。7.5 節も参照されたい。

2.2.1 公開鍵暗号の推定セキュリティ強度

CRYPTREC 暗号リストに掲載されている公開鍵暗号については、アルゴリズムに依存して、数体篩法、指数計算法、ρ 法といった解法アルゴリズムによる攻撃が最も効果的な攻撃方法である。そこで、これらの攻撃方法に基づいて推定される公開鍵暗号のセキュリティ強度をビットセキュリティで表現する。

---

[7] 1 つの候補が正しい秘密鍵であるかを判定するために必要な計算量を 1 として、どの程度の候補数を調べれば正しい秘密鍵を確実に（又は高い確率で）求められるかを表した値である。

表 3　公開鍵暗号の推定セキュリティ強度

セキュリティ強度 （ビットセキュリティ）	IFC RSA-PSS RSASSA-PKCS1-v1.5 RSA-OAEP RSAES-PKCS1-v1_5	FFC DSA DH	ECC ECDSA ECDH PSEC-KEM
112	k = 2048	(L, N) = (2048, 224)	P-224 B-233 K-233
128	k = 3072	(L, N) = (3072, 256)	P-256 B-283 K-283 W-25519 Curve25519 Edwards25519
192	k = 7680	(L, N) = (7680, 384)	P-384 B-409 K-409 W-448 Curve448 Edwards448
256	k = 15360	(L, N) = (15360, 512)	P-521 B-571 K-571

※ P: curve over Prime fields（素体曲線）、B: curve over Binary fields（拡大体（バイナリ）曲線）、K: Koblitz-curve（コブリッツ曲線）、W: Weierstrass-curve（ワイエルシュトラス曲線）、Curve: Montgomery-curve（モンゴメリ曲線）、Edwards: Edwards-curve（エドワード曲線）

以上を踏まえ、それぞれのビットセキュリティに対応するアルゴリズムの鍵長を示したのが表 3 である。2 行目のアルゴリズム名は、CRYPTREC 暗号リストに掲載されている公開鍵暗号のアルゴリズムを示している。

- 2 列目は、素因数分解問題ベースの公開鍵暗号（IFC：Integer Factorization Cryptography）を使用する場合の 1 列目で示したビットセキュリティを提供する鍵長（パラメータ）を示す。k は鍵長である。
- 3 列目は、有限体上の離散対数問題ベースの公開鍵暗号（FFC：Finite Field Cryptography）を使用する場合の 1 列目で示したビットセキュリティを提供する鍵長（パラメータ）を示す。L は公開鍵の鍵長、N はプライベート鍵の鍵長である。

- 4 列目は、楕円曲線暗号（ECC：Elliptic Curve Cryptography）を使用する場合の 1 列目で示したビットセキュリティを提供する曲線（パラメータ）を示す。一般に数字部分が鍵長に相当する（ただし、数字部分が 25519 の場合には鍵長 255 ビットに相当する）。例えば、P-256 は鍵長 256 ビットの素体曲線、B-283 は鍵長 283 ビットの拡大体（バイナリ）曲線、Edwards25519 は鍵長 255 ビットのエドワード曲線であることを示す。

2.2.2 共通鍵暗号の推定セキュリティ強度

CRYPTREC 暗号リストの「電子政府推奨暗号リスト」及び「推奨候補暗号リスト」に掲載されている共通鍵暗号については、鍵全数探索法が最も効果的な攻撃方法であるため、鍵全数探索法を用いた時のセキュリティ強度をビットセキュリティで表現する。一方、「運用監視暗号リスト」に掲載されている共通鍵暗号については、鍵全数探索法よりも効果的な攻撃方法（ショートカット攻撃法）が存在することが分かっているため、ショートカット攻撃法を用いた時の推定セキュリティ強度をビットセキュリティで表現する。

以上を踏まえ、それぞれのビットセキュリティに対応するアルゴリズム（及び鍵長）を示したのが表 4 である。

- 2 列目は、1 列目で示したビットセキュリティを提供するブロック暗号のアルゴリズム（及び鍵長）を示す。
- 3 列目は、1 列目で示したビットセキュリティを提供するストリーム暗号のアルゴリズムを示す。
- ブロック暗号を利用する暗号利用モード及びメッセージ認証コードのビットセキュリティは、ベースとなるブロック暗号のアルゴリズム（及び鍵長）に準拠する。

表 4 共通鍵暗号の推定セキュリティ強度

セキュリティ強度（ビットセキュリティ）	ブロック暗号※	ストリーム暗号	認証暗号
112	3-key Triple DES	－	－
128	鍵長 128 ビットを利用する CRYPTREC 暗号リスト掲載のブロック暗号	KCipher-2 Enocoro-128v2 MUGI	－
192	鍵長 192 ビットを利用する CRYPTREC 暗号リスト掲載のブロック暗号		－
256	鍵長 256 ビットを利用する CRYPTREC 暗号リスト掲載のブロック暗号	MULTI-S01	ChaCha20-Poly1305

※ ブロック暗号のセキュリティ強度はブロック長にも依存[8]するため、ブロック暗号を選択する際にはブロック長も併せて**考慮すべきである**。

2.2.3 ハッシュ関数の推定セキュリティ強度

ハッシュ関数については、利用方法によって要求される特性が異なるため、どちらのセキュリティ強度の推定値を使うのかは利用用途に応じて慎重に**判断すべきである**。特に、署名のように衝突困難性[9]を必要とするアプリケーションで使う場合（衝突困難性に対するセキュリティ強度に依存するケース）と、メッセージ認証コード（HMAC）や鍵導出（KDF）などのように衝突困難性を必要としないアプリケーションで使う場合（原像計算困難性[10]に対するセキュリティ強度に依存するケース）とを分けて考える必要がある。

衝突困難性に対するセキュリティ強度については、CRYPTREC 暗号リストの「電子政府推奨暗号リスト」及び「推奨候補暗号リスト」に掲載されているハッシュ関数のいずれにおいてもバースデーパラドックスよりも効率的に衝突するメッセージ組を求める効果的な攻撃方法が見つかっていないため、バースデーパラドックスによる衝突困難性に対するセキュリティ強度をビットセキュリティで表現する。なお、「運用監視暗号リスト」に掲載されているハッシュ関数 SHA-1 と RIPEMD-160 は、ハッシュ長が 160 ビットであるため、衝突困難性に対して 80 ビット以下[11]のセキュリティ強度しかない。このため、表 5 には含まれていない。

原像計算困難性に対するセキュリティ強度については、CRYPTREC 暗号リストに掲載されているハッシュ関数のいずれもが全数探索法よりも効果的な攻撃方法が見つかっていないため、全数探索法を用いた時のセキュリティ強度をビットセキュリティで表現する。

以上を踏まえ、それぞれのハッシュ関数のビットセキュリティを表現したのが表 5 である。

- 2 列目は、衝突困難性に対するセキュリティ強度に依存するケースにおいて、1 列目で示したビットセキュリティを提供するハッシュ関数のアルゴリズムを示す。署名と組み合わせてハッシュ関数を使う場合は、この列を参照すること。
- 3 列目は、原像計算困難性に対するセキュリティ強度に依存するケースにおいて、1 列目で示したビットセキュリティを提供するハッシュ関数のアルゴリズムを示す。メッセージ認証コード（HMAC）や鍵導出（KDF）にハッシュ関数を使う場合は、この列を参照するこ

---

8 一般にブロック長が長いほどセキュリティ（暗号学的安全性）が向上する。特にブロック暗号を使ってメッセージ認証を行う場合はその影響が大きい。現在では、128 ビットのブロック長を使うアルゴリズムが一般的である。

9 衝突困難性とは、同じハッシュ値を生成する 2 つのメッセージを見つけることが困難である性質のことをいう。効果的な攻撃方法が見つかっていないハッシュ関数では、ハッシュ長に対するバースデーパラドックスを基にしたセキュリティ強度となり、具体的にはハッシュ長の半分の値で表現される。例えば、ハッシュ長が 256 ビットである場合、バースデーパラドックスを基にしたセキュリティ強度は 128 ビットセキュリティとなる。

10 原像計算困難性とは、与えられたハッシュ値を生成するメッセージを構築したり見つけたりすることが困難である性質のことをいう。効果的な攻撃方法が見つかっていないハッシュ関数では、ハッシュ長に対する全数探索を基にしたセキュリティ強度となり、具体的にはハッシュ長の値で表現される。

11 SHA-1 については、衝突困難性に対してバースデーパラドックスよりも効果的な攻撃方法が見つかっているため、衝突困難性に対するセキュリティ強度は 80 ビットセキュリティにも達しない。

と。なお、利用する鍵のエントロピーがそのビットセキュリティ以上のエントロピーを有していることを前提とする。

表 5　ハッシュ関数の推定セキュリティ強度

セキュリティ強度 （ビットセキュリティ）	衝突困難性に対するセキュリティ強度に依存するケース （署名と組み合わせて利用する場合）	原像計算困難性に対するセキュリティ強度に依存するケース （HMAC や KDF に使う場合）
112	－	－
128	SHA-256 SHA-512/256 SHA3-256 SHAKE128 SHAKE256（ハッシュ長 256 ビット）	SHAKE128 SHA-1※ RIPEMD-160※
192	SHA-384 SHA3-384 SHAKE256（ハッシュ長 384 ビット）	－
256	SHA-512 SHA3-512 SHAKE256（ハッシュ長 512 ビット）	SHA-1、RIPEMD-160 及び SHAKE128 を除く CRYPTREC 暗号リスト掲載のハッシュ関数全て
備考	※SHA-1 及び RIPEMD-160 は、112 ビットのセキュリティ強度に達しないので、記載していない	※SHA-1 及び RIPEMD-160 は、192 ビットのセキュリティ強度に達しないので、128 ビットセキュリティに置いている

2.3 暗号技術の組合せによるセキュリティ強度の考え方

システムやアプリケーションによっては、2.1 節に記載された暗号処理のいくつかを組み合わせて実現することが求められる。このような場合、異なる種類の暗号処理に対して異なる暗号技術のアルゴリズムと鍵を使用する（例えば、暗号化に AES を使用し、署名に RSA を使用する）やり方もあれば、同じアルゴリズムと同じ鍵、又は同じアルゴリズムと異なる鍵で使用する（例えば、AES を使用して暗号化とメッセージ認証を実行する）やり方もある。また、利用するアルゴリズムも複数のアルゴリズムから選択できる場合もある（例えば、鍵共有において、公開鍵暗号なら RSA、Diffie-Hellman（DH）、ECDH などから、共通鍵暗号ならブロック暗号のいずれかのアルゴリズムを使った鍵ラッピング法から選択できる）。

そのため、システムやアプリケーションによっては、異なるセキュリティ強度を有する複数のアルゴリズムと鍵長を組み合わせて実現されることも多い。このような場合、最終的なセキュリティ強度は、最も弱いセキュリティ強度である暗号技術のアルゴリズムと鍵長によって決定され

る[12]（図 2 参照）。

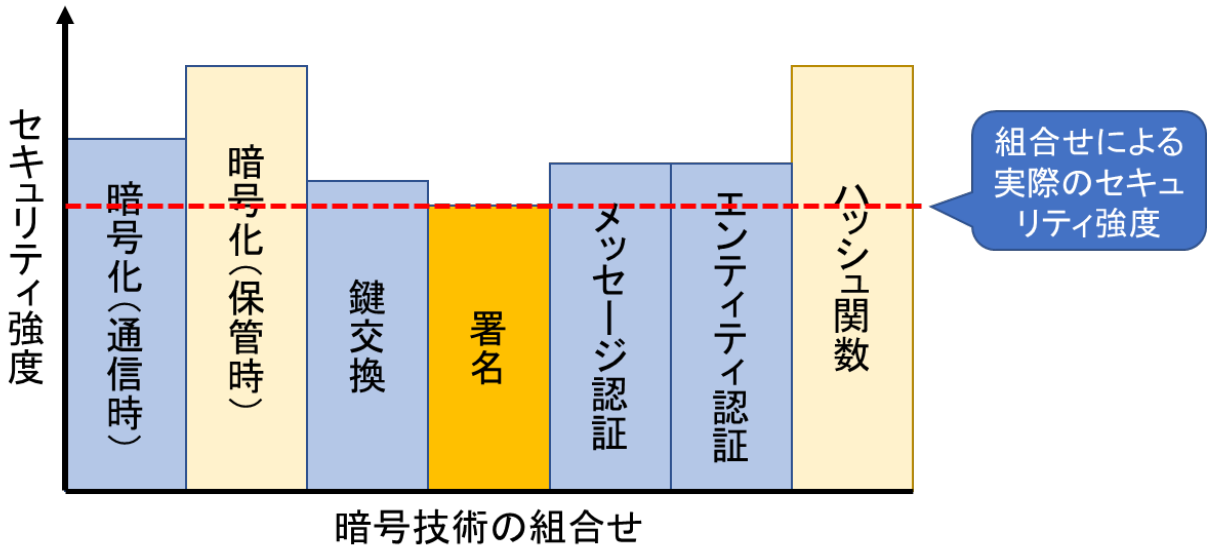

図 2　暗号技術の組合せによるセキュリティ強度（イメージ図）

以下に、いくつかの暗号技術の組合せ例を用いてセキュリティ強度の考え方を示す。

- 暗号通信において、セッション鍵の確立を公開鍵暗号で行い、データの暗号化は共通鍵暗号で行うハイブリット暗号化方式の場合、そのセキュリティ強度はより弱い方のアルゴリズムと鍵長によって決定される。例えば、256 ビット鍵の AES でデータの暗号化をする場合、通常であれば 256 ビットのセキュリティ強度を提供する。しかし、256 ビットのセッション鍵を確立するために P-256 ビット鍵（素体曲線での鍵長 256 ビットの鍵）の ECDH が使用される場合、P-256 ビット鍵の ECDH は 128 ビットセキュリティに該当するため（2.2.1 節参照）、そのセッション鍵で保護されたデータに対しては（256 ビットセキュリティではなく）128 ビットのセキュリティ強度しか提供されない。
- ハッシュ関数と署名アルゴリズムを組み合わせて署名を計算する場合、署名のセキュリティ強度はより弱い方のアルゴリズムによって決定される。例えば、SHA-256 を 2048 ビット鍵の RSA 署名と組み合わせて使用する場合、2048 ビット鍵の RSA 署名は 112 ビットセキュリティに該当するため（2.2.1 節参照）、その署名に対して（128 ビットセキュリティではなく）112 ビットのセキュリティ強度しか提供されない。

所定のセキュリティ強度をサポートするためには、アルゴリズムと鍵長の組合せを慎重に**選択すべきである**。例えば、通信されるデータを保護するために 128 ビットセキュリティ強度で暗号化、署名及び鍵共有を行う場合、以下のような暗号技術の選択の組合せが考えられる。

12 「樽理論」等とも呼ばれる。

i) 暗号化：共通鍵暗号で 128 ビットセキュリティ強度を有するアルゴリズム（と鍵長）のなかから選択する（例えば、128 ビット鍵の AES）。

ii) 署名：SHA-256 を署名生成前のデータハッシュに使用する。署名アルゴリズムは、128 ビットセキュリティ強度を有するアルゴリズムと鍵長の組合せのなかから選択する（例えば、3072 ビット鍵の RSA 署名）。なお、同一のビットセキュリティ強度で複数のアルゴリズムと鍵長が利用可能な場合、アルゴリズムの性能、メモリ要件などに基づいて選択してよい。

iii) 鍵共有：128 ビットセキュリティ強度を有するアルゴリズムと鍵長の組合せのなかから選択する。例えば、ECDH が利用可能な場合は、ECDH と 128 ビットセキュリティ強度の楕円曲線（P-256 など）の組合せを使用する。

3. 鍵長選択の考え方

本節では、保護対象のデータに対してシステムやアプリケーションが適切な保護を提供するための鍵長選択の考え方を提示する。鍵長の選択にあたっては利用する暗号処理の種類に依存することにも留意されたい。

3.1 運用寿命とセキュリティ強度要件の関係

システムやアプリケーションを設計・開発する際は、そのシステムやアプリケーションの検討・設計開始から構築、運用、さらに運用終了・廃棄までの運用寿命全体と、その期間に実現するセキュリティ強度の関係を考慮して適切なセキュリティ強度要件を設定し、その強度要件を満たす鍵長を**選択すべきである**。これは、時間の経過とともに解読計算能力が向上するため、運用開始時と比較して安全性が低下し、攻撃が成功する可能性が高まるリスクがあるためである。

結果として、システムやアプリケーションの運用途中でより安全な鍵長（又はより安全なアルゴリズム）への移行が必要となる場合があることにも留意されたい。また、システムやアプリケーションの運用中における予期しない危殆化等への対処のため、鍵長を容易に変更できるように配慮した移行計画を考慮するのが望ましく（7 節参照）、特に運用寿命が長期にわたるシステムやアプリケーションの場合には重要な視点である。

【重要な注意】

1.1 節に記載の通り、量子コンピュータによる暗号技術の危殆化は将来的なリスク要因に位置づけている。そのため、運用寿命が長期にわたるシステムやアプリケーションであって、特にその中で公開鍵暗号や署名を利用している場合には、将来的に耐量子計算機暗号（PQC: Post-Quantum Cryptography）の採用も視野に入れた移行計画が必要となる場合があることに留意されたい（7.5 節参照）。

本書では、システムやアプリケーションの運用寿命の期間と求められるセキュリティ強度要件の関係から 3 つの要件設定方法を示す（図 3 参照）。システムやアプリケーションの検討状況を踏まえ、適切な要件設定方法を選択されたい（図 4 参照）。

【要件設定方法①】

システムやアプリケーションの運用寿命全体を通して必要なセキュリティ強度要件を設定し、その強度要件を満たす鍵長を**サポート（実装）すべきである**。その際、必要なセキュリティ強度を過小評価又は過大評価しないように**注意すべきである**。

なお、利用終了時期を明確化し、それまでにより安全な鍵長に移行することを条件に、その期間中は安全と期待される鍵長を一緒にサポート（実装）してもよい。

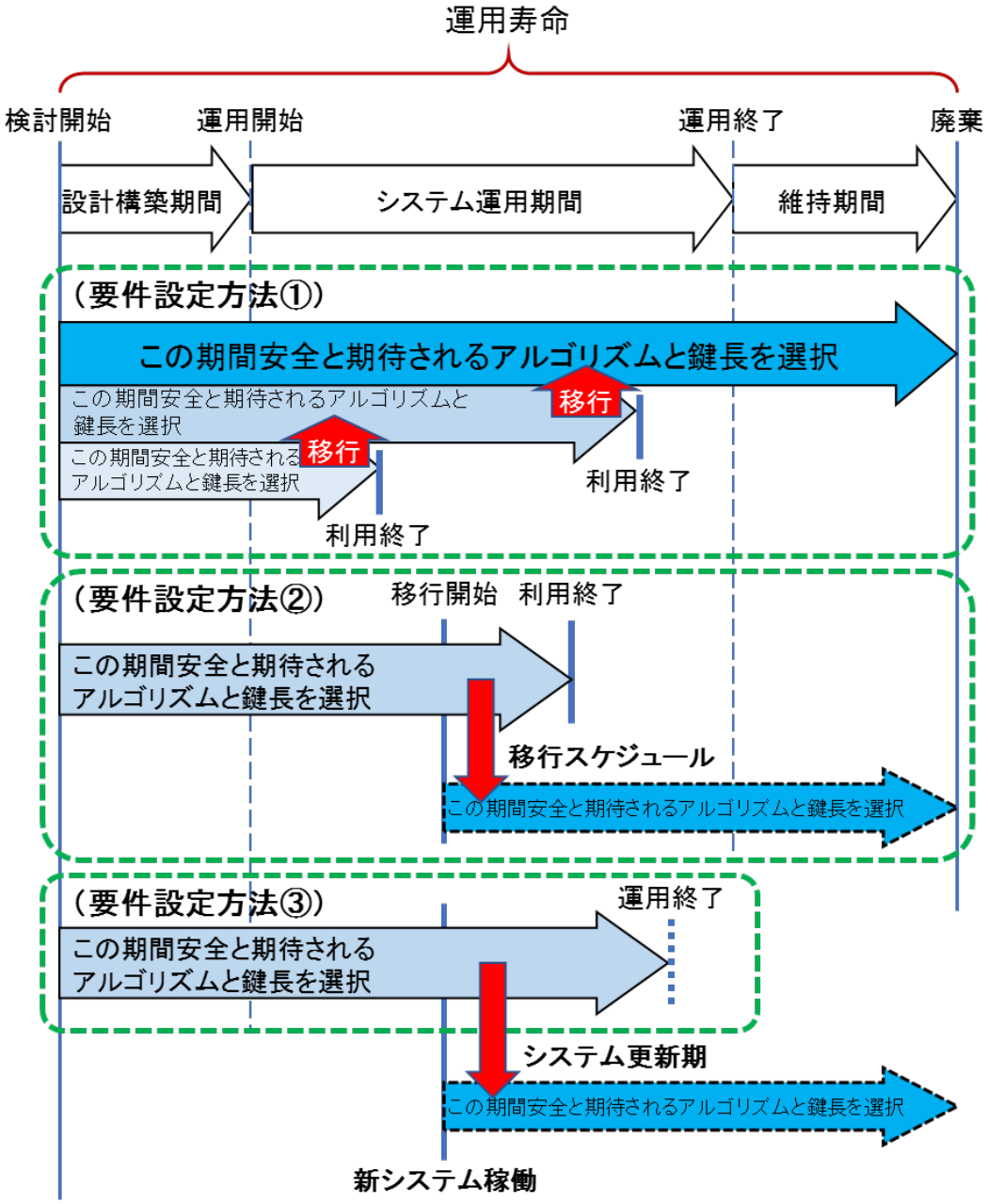

図 3　システムの運用寿命と求められるセキュリティ強度要件

【要件設定方法②】

対象となるシステムやアプリケーションの設計・開発における何らかの制約により、運用寿命全体を通して必要なセキュリティ強度要件を当初から設定することが困難である場合には、セキュリティ強度要件を切り替える移行時期を明確化したスケジュールを立案したうえで、その移行時期終了まで安全に運用するために必要なセキュリティ強度要件を設定し、その強度要件を満たす鍵長を**サポート（実装）すべきである**。その際、そのスケジュールには移行開始予定時期及び移行完了予定時期を明示することが望ましい。

【要件設定方法③】

対象となるシステムやアプリケーションにおいて、運用寿命が決まっていない（明確ではない）場合には、システムやアプリケーションの更新期を明確化したスケジュールを立案したうえで、その更新期まで安全に運用するために必要なセキュリティ強度要件を設定し、その強度要件を満たす鍵長を**サポート（実装）すべきである**。なお、そのスケジュールにおいて、新システムや新アプリケーションの稼働開始予定時期及び併用運用想定期間を示しておくことが望ましい。

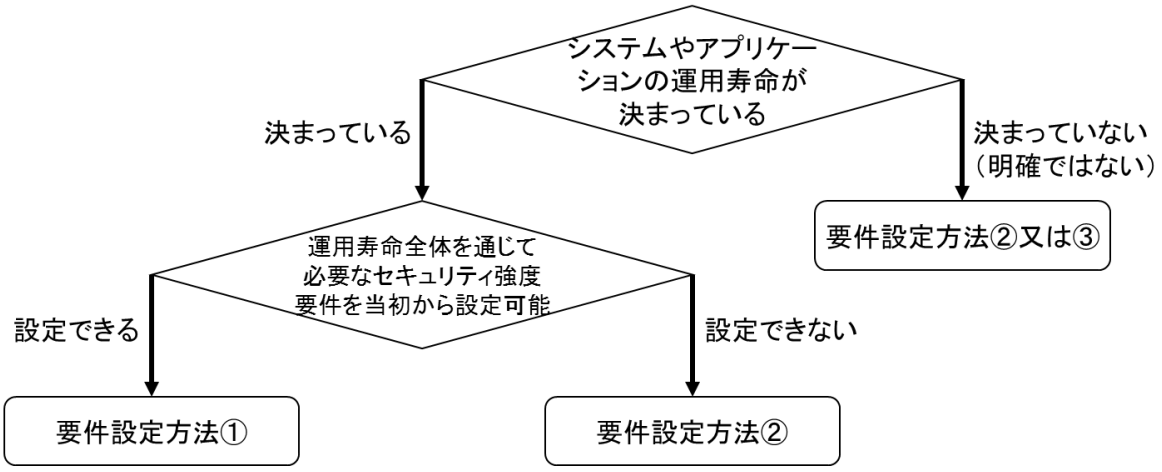

図 4　要件設定方法の選択フロー

3.2 求められるセキュリティ強度要件の考え方

必要なセキュリティ強度を設定する場合、システムやアプリケーションの運用開始年ではなく、予定している運用終了・廃棄年又は利用期間の終了年を基準として、表 6 を参考に考えることが望ましい。例えば、システムの運用終了・廃棄年を 2037 年に予定しているであれば「2040」の列を、2053 年に予定しているのであれば「2060」の列を参照してセキュリティ強度を設定する。システムの運用開始年が 2023 年であっても「2030」の列を参照するわけではない。

表 6 は 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表している。これは、解読能力の向上が計算機能力の向上だけにほぼ比例する（いわゆるムーアの法則[13]に従って向上していく）と仮定した場合にその後 10～15 年程度であれば解読が困難を期待される状態であることを意味する。ただし、未知の攻撃手法に対するセキュリティ（暗号学的安全性）の余裕度（“セキュリティマージン”という）がほとんどない状態なので、画期的な新たな解読手法が見つかるなどして解読能力が急激に向上した場合には、10～15 年持たずに解読される可能性があることに留意されたい。

[13] 「集積回路上のトランジスタ数が 18 ヶ月ごとに 2 倍になる」という経験法則のこと

したがって、**表 6 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。**例えば、「2040」の列であれば最低で 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。

このほかに、「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」や Appendix に記載の情報を参照してセキュリティ強度を設定してもよい。

表 6　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）[14]	56	81 ～ 96	86 ～ 104	91 ～ 112	96 ～ 120	101 ～ 128
Lenstra（2001）[15]	56	93	101	109	－	－
Lenstra（2004）[16]	56	88	95	102	－	－

ちなみに、CRYPTREC では、「素因数分解の困難性に関する計算量評価」と「楕円曲線上の離散対数計算の困難性に関する計算量評価」の予測図を作成[17]している（Appendix 図 7 参照）。これは、公開鍵暗号の鍵長の選択について検討するために、危殆化の様子を分かり易く示すために作成されたものである。

この予測図では、世界最速及び世界 500 位のスーパーコンピュータの計算機能力がムーアの法則に従って向上していくと仮定して外挿線を直線で引いたものである。極めて画期的な暗号解読手法が発明され、全く想定外のセキュリティ強度の低下が生じるような事象が生じない限り、この直線と所定のセキュリティ強度が交わる時期に解読可能になると見込まれるため、信頼性の高い下限のセキュリティ強度を表す指標として活用できる。

3.3 鍵長の選択及び利用期間の考え方

システムやアプリケーションの設計・開発においては、3.2 節を踏まえて設定したセキュリティ強度と同じかそれ以上のセキュリティ強度を満たす鍵長を選択して**サポート（実装）すべきである**。例えば、必要なセキュリティ強度として 128 ビットセキュリティが設定された時、公開鍵

[14] Mécanismes cryptographiques - Règles et recommandations, Rev. 2.03, ANSSI , 02/2014 https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B1.pdf

[15] Selecting Cryptographic Key Sizes, Arjen K. Lenstra and Eric R. Verheul, Journal of Cryptology, vol. 14, p. 255-293, 2001. https://link.springer.com/content/pdf/10.1007/s00145-001-0009-4.pdf

[16] Key Lengths, Arjen K. Lenstra, The Handbook of Information Security, 06/2004. http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.694.8206

[17] CRYPTREC Report 2021 暗号技術評価委員会報告、https://www.cryptrec.go.jp/report/cryptrec-rp-2000-2021.pdf

暗号であれば鍵長 3072 ビットの RSA（2.2.1 節表 3 参照）、共通鍵暗号であれば鍵長 128 ビットの CRYPTREC 暗号リスト掲載のブロック暗号(2.2.2 節表 4 参照)、ハッシュ関数であれば SHA-256（2.2.3 節表 5 参照）などが選択肢となる。

なお、設定したセキュリティ強度以下のセキュリティ（暗号学的安全性）の鍵長をサポート（実装）してもよい。ただし、サポート（実装）された鍵長の全てが常に利用されてよいわけではないことに**留意すべきである**。サポート（実装）された鍵長の利用期間については、そのセキュリティ強度に応じて適切に**定めるべきである**。

具体的には、特定の鍵長について、保護されたデータが安全であり続けると評価された期間は「当該鍵長のセキュリティ寿命」と呼ばれ、その期間中はどの対象データに対しても適切な保護を提供することが期待される。一方、特定のデータに対して暗号保護が適用されてから最終的に処理をする必要がなくなるまでの期間（つまり、機密性や完全性を保持する必要がある期間）は「当該データのセキュリティ寿命」と呼ばれ、その期間中は当該データに対して適切な保護を提供することが期待される。なお、データのセキュリティ寿命は暗号処理の違いにより異なる（表 7 参照）。

一般に「鍵長のセキュリティ寿命」と「データのセキュリティ寿命」の間には、以下のような関係性が成り立つように**設定すべきである**（図 5 参照）。特に、「データのセキュリティ寿命」は利用する鍵の「鍵長のセキュリティ寿命」に包含されるように**扱うべきであり**、包含されないような場合には、鍵長の移行を**検討すべきである**（7 節参照）。

「新規データに対する暗号保護の適用期間 [18]」＋「データのセキュリティ寿命」
≦「保護されたデータに対する処理の適用期間 [19]」
≦「鍵長のセキュリティ寿命」

表 7　データのセキュリティ寿命

暗号化	通信時	（鍵共有後から）通信が終了するまでの期間
	保管時	データが暗号化されてから、復号する必要がなくなるまでの期間
	鍵共有	鍵共有を行っている期間
署名		データに署名生成してから、当該署名の署名検証を行う必要がなくなるまでの期間。多くの場合、法令や規則などで定められる期間、又は署名検証用の公開鍵証明書の有効期間に依存する
メッセージ認証		データに対するメッセージ認証コード（MAC）を生成してから、当該データの完全性検証を行う必要がなくなるまでの期間
エンティティ認証		エンティティ認証を行っている期間

18 データ暗号化対称鍵（暗号化）、鍵共有対称鍵（暗号化）、鍵共有公開鍵、署名プライベート鍵、認証対称鍵（MAC 生成）、認証プライベート鍵の利用期間に一致する

19 データ暗号化対称鍵（復号）、鍵共有対称鍵（復号）、鍵共有プライベート鍵、署名検証公開鍵、認証対称鍵（MAC 検証）、認証公開鍵の利用期間に一致する

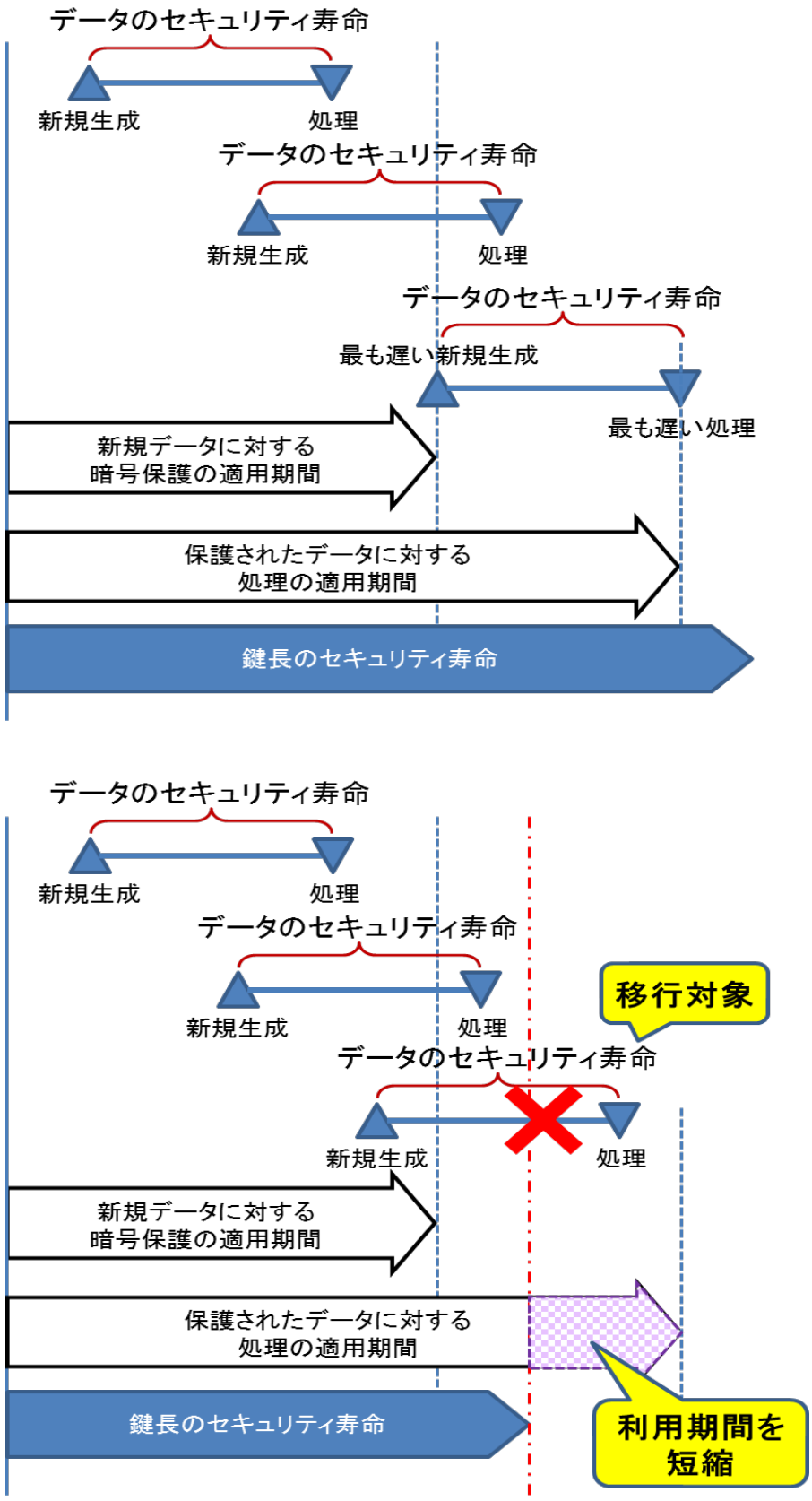

図 5　鍵長のセキュリティ寿命とデータのセキュリティ寿命の関係性

4. 鍵のライフサイクル

暗号で利用する鍵は永遠に使えるわけではない。図 6 は、NIST SP800-57 パート 1 改訂 5 版に記載されている、暗号鍵の生成から破棄までのライフサイクルにおける鍵状態とその間の遷移を示したものである。暗号鍵の取扱いで重要なのは、利用する様々な鍵に対して、この鍵のライフサイクルを正しく運用管理することである。

なお、「鍵の利用期間」とは当該鍵が「活性化状態」（一部は「非活性化状態」を含む）に存在する期間のことである。詳しくは 5 節を参照されたい。

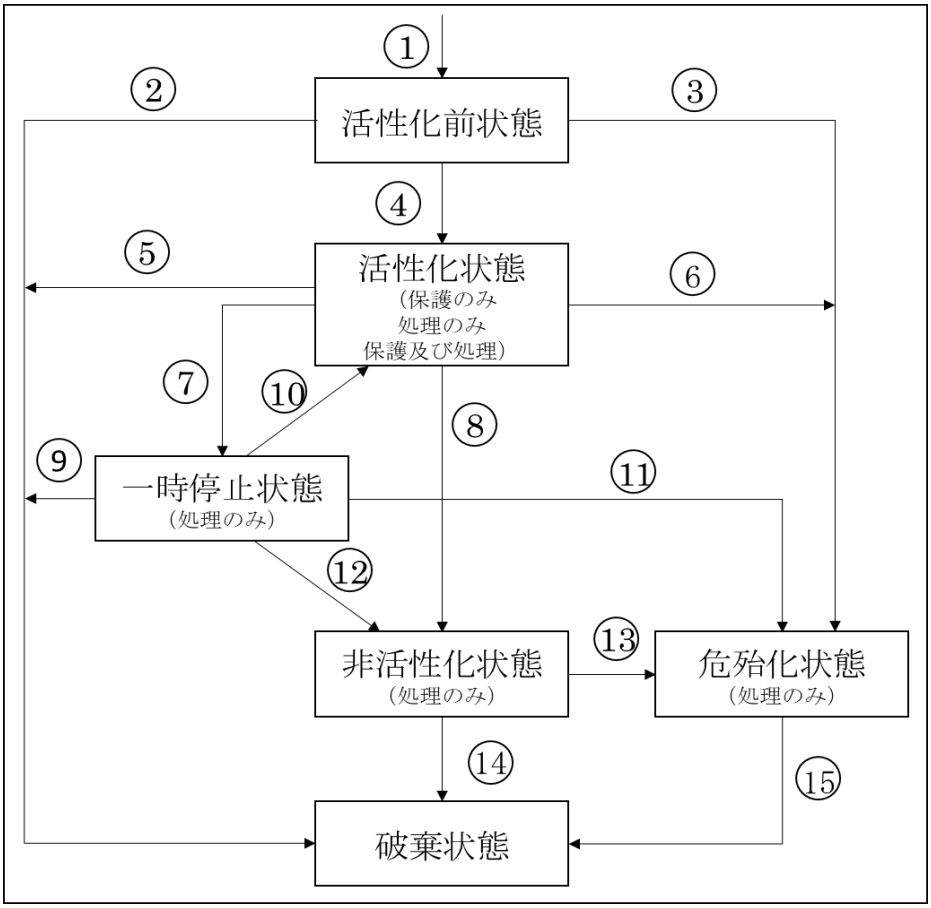

図 6　鍵状態と遷移

4.1 活性化前状態

鍵は生成されたが、使用開始前の状態である。この段階では、新規データに対して暗号保護の適用（暗号化や署名生成など）を行ったり、保護されたデータの処理（復号や署名検証など）を**行ってはならない**。

状態遷移①：

- 鍵は、生成された時点で直ちに活性化前状態に入る。

状態遷移②：

- 鍵が活性化前状態にあり、将来的にその鍵が必要ないと判断された場合、当該鍵は、活性化前状態から破棄状態に直接**遷移しなければならない**。
  なお、公開鍵暗号又は署名の場合には、鍵ペア（公開鍵とプライベート鍵の組）の両方の鍵が破棄状態に**遷移しなければならない**。

状態遷移③：

- 鍵が活性化前状態にあり、機密性保護が必要な鍵の機密性、又は鍵の完全性が疑われる場合には、当該鍵は、活性化前状態から危殆化状態に**遷移しなければならない**。
  なお、公開鍵暗号又は署名の場合には、鍵ペア（公開鍵とプライベート鍵の組）の両方の鍵が危殆化状態に**遷移しなければならない**。
- 危殆化状態に遷移する鍵が複数のエンティティに知られている場合、失効通知が**行われなければならない**。

状態遷移④：

- 対称鍵の利用期間、又は公開鍵とプライベート鍵の両方の利用期間が始まると、活性化前状態から活性化状態に**遷移しなければならない**。この遷移は、活性化する日付になった時に実行されてもよいし、外部イベントにより実行されてもよい。鍵がすぐに使用できるように生成された場合は、活性化前状態に入った直後に遷移が行われる。
- 証明書に関連付けられた鍵ペア（公開鍵とプライベート鍵の組）の場合、当該鍵ペアの公開鍵に対して最初に発行された証明書内に記載される *notBefore* の日付になったら、鍵ペアの両方の鍵が活性化状態に遷移する。

4.2 活性化状態

鍵が、新規データに対する暗号保護の適用、保護されたデータの処理、又はその両方に使用される状態である。なお、暗号保護の適用に利用できる期間は、活性化状態にある場合に限られる（5 節参照）。

状態遷移⑤：

- 鍵共有公開鍵、署名プライベート鍵、及び認証プライベート鍵と認証公開鍵については、鍵が危殆化することなく、その鍵の利用期間が終了した場合、活性化状態から破棄状態に直接**遷移しなければならない**。なお、対応する鍵共有プライベート鍵及び署名検証公開鍵は、この時点で活性化状態から非活性化状態に遷移することに留意されたい。

状態遷移⑥：

- 対称鍵又はプライベート鍵の危殆化が疑われた場合、又はそれが確認された場合、当該鍵（対応する公開鍵も）は活性化状態から危殆化状態に**遷移しなければならない**。
- 危殆化状態に遷移する鍵が複数のエンティティに知られている場合、失効通知が**行われなければならない**。

状態遷移⑦：

- 何らかの理由により、対称鍵又はプライベート鍵が一定期間使用されないのであれば、活性化状態から一時停止状態に**遷移しなければならない**。例えば、署名プライベート鍵は、当該鍵に関連付けられたエンティティが長期休暇中である、又はその鍵が危殆化した疑いがあるといった理由で、一時停止されることがある。後者の場合、失効及び交換のプロセスを開始する前に、鍵の状態を調査することができる。
  なお、公開鍵暗号又は署名の場合には、公開鍵とプライベート鍵の両方を同時に活性化状態から一時停止状態に**遷移しなければならない**。
- 一時停止状態に遷移する鍵が複数のエンティティに知られている場合、一時停止とその理由を示す通知が**行われなければならない**。

状態遷移⑧：

- 対称鍵については、新規データに対して暗号保護を適用する必要がなくなった場合、活性化状態から非活性化状態に**遷移しなければならない**。
- 鍵共有公開鍵又は署名プライベート鍵が活性化状態から破棄状態に直接遷移した場合、対応する鍵共有プライベート鍵及び署名検証公開鍵は、この時点で活性化状態から非活性化状態に**遷移しなければならない**。

4.3 一時停止状態

鍵の使用を一時的に中断させる状態である。この状態にある間は、新規データに対する暗号保護の適用を**行ってはならない**。

一時停止の理由には大きく 2 つある。一つは、鍵の危殆化が疑われる場合、当該鍵の失効及び交換のプロセスを開始する前に、状況を調査するための時間を確保する目的で行われるケースであり、もう一つは、鍵を所有するエンティティが当該鍵を利用できない場合（例えば、長期休暇中）に当人に無断で新規データに対する暗号保護の適用や保護されたデータの処理が行われないようにするためのケースである。

一時停止された鍵は、理由に応じて、活性化状態、非活性化状態、破棄状態、又は危殆化状態に遷移しうる。

状態遷移⑨：

- 一時停止状態中に利用期間が終了した鍵は、一時停止状態から破棄状態に**遷移しなければならない**。

状態遷移⑩：

- 鍵が危殆化しておらず、一時停止の理由が存在しなくなり、且つ暗号保護の適用の利用期間が残っている場合、一時停止状態にある鍵は、一時停止状態から活性化状態に**遷移しなければならない**。
  なお、公開鍵暗号又は署名の場合には、公開鍵とプライベート鍵の両方を同時に一時停止状態から活性化状態に**遷移しなければならない**。

状態遷移⑪：

- 対称鍵又はプライベート鍵の危殆化が疑われた場合、又はそれが確認された場合、当該鍵（対応する公開鍵も）は一時停止状態から危殆化状態に**遷移しなければならない**。
- 危殆化状態に遷移する鍵が複数のエンティティに知られている場合、失効通知が**行われなければならない**。

状態遷移⑫：

- 鍵が危殆化しておらず、一時停止の理由が存在しなくなった場合であって、新規データに対する暗号保護の適用の利用期間は残っていないが、保護されたデータの処理の利用期間が残っている場合、一時停止状態から非活性化状態に**遷移しなければならない**。一般に、データ暗号化対称鍵、認証対称鍵、及び署名検証公開鍵が対象となり得る。

4.4 非活性化状態

鍵が、新規データに対する暗号保護を適用するために**使用されてはならない**が、保護されたデータを処理するために使用できる状態である。一般に、データ暗号化対称鍵、認証対称鍵、及び署名検証公開鍵が対象となり得る。

状態遷移⑬：

- データ暗号化対称鍵又は認証対称鍵の危殆化が疑われた場合、又はそれが確認された場合、当該鍵は非活性化状態から危殆化状態に**遷移しなければならない**。
- 危殆化状態に遷移する鍵が複数のエンティティに知られている場合、失効通知が**行われなければならない**。

状態遷移⑭：

- 保護されたデータの処理に対する鍵の利用期間が終了した場合、又は保護されたデータを処理することが不要になった場合、非活性化状態から破棄状態に**遷移しなければならない**。

4.5 危殆化状態

対称鍵又はプライベート鍵の危殆化が疑われたり、確認されたりした状態である。危殆化した

鍵は、新規データに対する暗号保護を適用するために**使用されてはならない**。

一方、場合によっては、セキュリティ（暗号学的安全性）が担保されない可能性があるというリスクを受容し、高度に管理された条件下[20]において、保護されたデータを処理するために使用することがある。例えば、危殆化する前に生成された署名であり、署名後はずっと物理的に保護されていた場合、又は信頼できるタイムスタンプが署名データに含まれている場合であれば、署名データの完全性を判断するために検証されてもよい。

このように、危殆化した鍵の継続使用は、既に保護されているデータの処理に**限定されなければならず**、またその処理の結果に対して包含される危険性を十分に**認識すべきである**。

状態遷移⑮：

- 必要とされなくなったりした時点で、危殆化状態から破棄状態に**遷移すべきである**。

4.6 破棄状態

この状態の鍵は、いかなる暗号処理にも**使用されてはならない**。

対称鍵及びプライベート鍵は、コピーを含めて、当該鍵の痕跡を全て除去する方法で破棄して、物理的又は電子的な手段では**復元できない**[21]**ようにすべきである**（NIST SP800-88 改訂 1 版[22]参照）。公開鍵は、必要に応じて保持又は破棄することができる。

20 例えば、情報の機密性や完全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のことを指す。

21 単純な削除では、鍵を完全には消去できない場合があることに留意されたい。

22 NIST SP800-88 Revision1 の日本語訳、https://www.ipa.go.jp/files/000094547.pdf

5. 鍵タイプごとの鍵の利用期間

5.1 鍵の利用期間

利用する鍵は鍵のライフサイクルを踏まえて運用することが必要である（4 節参照）。

鍵の利用期間とは、その鍵に対して正規のエンティティが使用することを認められている期間、又はあるシステムやアプリケーションにおいてその鍵が有効である期間のことである。すなわち、4 節図 6 の*活性化状態*に存在することができる期間（データ暗号化対称鍵、認証対称鍵、及び署名検証公開鍵については*非活性化状態*の期間も含む）のことである。

鍵の利用期間の長さと、鍵の危殆化リスクやインシデントに伴う影響度とはトレードオフの関係にあるので、利用期間は慎重に検討して適切に**決定すべきである**。なお、鍵が危殆化した場合、それ以降の利用期間は有効と**見なすべきではない**。

また、以下の理由により、暗号処理ごとに異なる鍵を**使用すべきである**。

1. 鍵が危殆化した場合に被害が発生する可能性のある範囲を限定する
2. 2 つ以上の異なる暗号処理に同じ鍵を使用すると、一部又は全ての暗号処理で提供されるセキュリティ強度が低下する可能性がある
3. 暗号処理の違いにより、鍵管理上の要件が干渉することがある。例えば、署名プライベート鍵は署名生成が許可されている期間が経過したら直ちに**破棄すべきである**が、鍵共有プライベート鍵は、鍵共有が続く限り、破棄できない。

5.2 鍵の利用期間に影響を与える要因

一般に、鍵の利用期間が短いほど、その鍵に対するセキュリティ（暗号学的安全性）が向上する。しかしその一方で、鍵更新を頻繁に行うことが必要になるので、鍵の更新方法によっては人為的なエラーや設定ミスなどによる鍵の漏えいリスクが高まる可能性も想定し得る。

また、ビットセキュリティ強度のより高い鍵長を選択すれば、将来的な危殆化に対する耐性が向上すると期待されるので、そのアルゴリズムで使う鍵の利用期間を延ばすことが容易になる。ただし、その場合は、暗号解読による攻撃リスクよりも、システムへの侵入や破壊など、暗号解読以外の攻撃方法によって、より少ない時間とコストで鍵に直接アクセスされるリスクのほうが大きくなる可能性があることに**留意すべきである**。このため、鍵の保護が非常に重要になる（6 節参照）。

このように、鍵の利用期間は、当該鍵が使用されるシステムやアプリケーションの状況、利用環境等に大きく影響されるので、それらの状態に応じて適切に**設定されるべきである**。鍵の利用期間の長さに影響を与える要因の中には、以下のようなものがある（これらに限るわけではない）。

1. 暗号処理の種類

2. アルゴリズムのセキュリティ強度（例：鍵長、ブロック長など）
3. アルゴリズムの使用に必要な制限（例：ノンスの再利用を避けるための最大呼び出し回数）
4. 動作環境（アクセスが限定された安全な施設か、オープンオフィス環境か、一般にアクセス可能な端末や攻撃者が入手可能なデバイスか、など）
5. 同一の鍵で処理するデータ量又はトランザクション数（例：同一の鍵で暗号化された平文と暗号文のペア数など）
6. データのセキュリティ寿命
7. 秘密に保持する鍵の保護方法（例：物理的又は論理的なアクセス制御やセキュリティ対策など）及び保護環境（例：耐タンパーモジュール（TPM、HSM、SIM など）の使用など）
8. 秘密に保持する鍵のコピー数、及びそのコピーの配付先（例：鍵のバックアップ／アーカイブ、共有するエンティティなど）
9. 想定攻撃者の能力（例：想定される技術的攻撃能力や資金力など）
10. 新技術（例：量子コンピュータなど）による暗号解読成功確率の向上
11. 鍵が漏えいした場合や保護されている情報の機密性が失われた場合に想定されるインパクト（例えば、不正なエンティティへの情報漏えい、署名の偽造など）
12. 鍵の更新方法（例：自動更新か手動更新か、など）
13. 鍵の更新に関連するコスト（例：大規模データベースや分散型データベースの再暗号化、一度に非常に多くの鍵の交換が必要になるケースなど）

5.3 鍵タイプごとの鍵の利用期間の考え方

5.3.1 公開鍵暗号及び署名の鍵ペアの利用期間

公開鍵暗号及び署名の鍵ペア（公開鍵とプライベート鍵の組）では、ペアの鍵ごとに自身の利用期間を独立して設定する。鍵ペアの一方の鍵は、新規データに対する暗号保護の適用に使用され、もう一方の鍵は保護されたデータの処理に使用される。

- 鍵共有での鍵ペア（鍵共有公開鍵と鍵共有プライベート鍵）では、鍵配送スキーム[23]を使う場合、通常、鍵を配送するために暗号化する鍵共有公開鍵の利用期間よりも、鍵を復号する鍵共有プライベート鍵の利用期間のほうが長くなる。鍵合意スキーム[24]を使う場合、両方の鍵の利用期間は通常同じである。
- 署名での鍵ペア（署名プライベート鍵と署名検証公開鍵）の場合、通常、署名生成をするための署名プライベート鍵の利用期間よりも、署名検証をするための署名検証公開鍵の利用期間のほうが長くなる。なお、署名検証公開鍵が公開鍵証明書で配付される場合、その鍵の利用期間は当該公開鍵証明書の *notBefore* と *notAfter* の日付で示される期間（つまり、当該公開鍵証明書の有効期間）となる。また、署名生成できる利用期間が経過した場合、当該署名プライベート鍵は**破棄されなければならない**。

[23] 鍵生成者が送信者となり、受信者に当該鍵を送付する一方向での鍵共有になる
[24] 双方が鍵導出に必要な情報を相互に送付し、各自が鍵導出することで秘密鍵を共有する鍵共有になる

- 認証での鍵ペア（認証プライベート鍵と認証公開鍵）の場合、多くは、両方の鍵の利用期間は同じである。通常、認証プライベート鍵がチャレンジ情報の署名に使用されなくなった場合には、認証公開鍵も不要となる。

5.3.2 共通鍵暗号の鍵の利用期間

共通鍵暗号では、新規データに対する暗号保護の適用と保護されたデータの処理の両方で同じ対称鍵が使われる。

- 通信の暗号化に使用される場合、発信者がデータを暗号化してから受信者が復号するまでの時間は比較的短いと考えられる。この場合、暗号化に使える利用期間と復号に使える利用期間とは通常同じである。
  なお、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、Retrospective Decryption ともいう）を強く想定する必要がある場合には、通信時だけを考慮するのではなく、初めから保管を想定した暗号化の方法を**準用すべきである**。この攻撃に対しては後から防ぐことができないため、こういった攻撃に対する対策が必要であるかどうかについても、予め十分に**検討すべきである**。
- 保管データの機密性（暗号化）又は完全性（メッセージ認証）を保護するために使用される場合、通常、新規データに対して暗号保護に適用するための利用期間よりも、保護されたデータを処理するための利用期間のほうが長くなる。しかし、同じ対称鍵を使うことが必要となる関係上、新規データに対する暗号保護の適用のための利用期間を経過した後は、新たな暗号保護に適用しないように、運用による対策を**立てるべきである**。
- エンティティ認証のために使用される場合、被認証者がチャレンジ情報を暗号化してから検証者が復号して当該チャレンジ情報を得るまでの時間は比較的短いと考えられる。この場合、暗号化に使える利用期間と復号に使える利用期間とは通常同じである。

5.3.3 SP800-57 に記載されている推奨利用期間

様々な鍵タイプに対する推奨利用期間が NIST SP800-57 パート 1 改訂 5 版に記載されており、表 8 に示す。なお、この推奨利用期間の多くは、米国政府での運用効率を最大化したいという要望と、使用環境の最低基準を想定したものを基準としている。

したがって、この内容はあくまでも大まかなガイダンスであると**理解すべきであり**、この通りにすることを要求しているわけではない。実際には当該鍵が使用されるシステムやアプリケーション、利用環境等に依存して、より長い利用期間やより短い利用期間を設定することが適切とされる場合があることに注意されたい。

5.2 節の要因を踏まえ、利用期間の長さを適切に**設定すべきである**。

表 8　鍵タイプごとの SP800-57 に記載されている推奨利用期間

鍵タイプ	推奨利用期間	
	新規生成	処理
データ暗号化対称鍵	2 年以内	新規生成の利用期間終了後、3 年以内
鍵共有対称鍵	2 年以内	新規生成の利用期間終了後、3 年以内
鍵共有プライベート鍵	―	2 年以内
鍵共有公開鍵	1～2 年	―
署名プライベート鍵	1～3 年	–
署名検証公開鍵	―	数年（鍵長に依存）
認証対称鍵	2 年以内	新規生成の利用期間終了後、3 年以内
認証プライベート鍵	1～2 年	―
認証公開鍵	―	対応する認証プライベート鍵の利用期間と同じ

6. 鍵の保護について

6.1 鍵の保護要件

鍵は、対象とする暗号処理で使う必要がある限り、適切に利用できることが求められる。そのために当該鍵が満たすべき保護要件として考慮すべき項目は以下の通りである。ただし、表 9 に示すように、必要な保護要件は鍵タイプによって異なることに留意されたい。

セキュリティ特性

鍵に要求されるセキュリティ特性（すなわち、機密性や完全性、可用性）のことである。

- 機密性は、秘密にしておくことを意図した全ての鍵に要求される。一般に、適切なセキュリティレベルの CMVP[25]認証済暗号モジュールの利用、適切なセキュリティ強度での暗号化の実施、又は秘密情報への適切なアクセス制御下にある領域での保管などによって実現される。
- 完全性は、全ての鍵に要求される。
- 可用性は、保護されたデータの処理（復号や検証など）を行う際に必要となる全ての鍵に要求される。これは、鍵のコピーやバックアップ、アーカイブなどによって実現する場合もある。

関連性保護

ある鍵を使って暗号処理を正しく実行する際に、(当該鍵以外に) 適正に保護されていることが求められる鍵やデータのことを表す。

例えば、署名検証公開鍵を使って署名を検証しようとする際には、当該鍵に対応する署名プライベート鍵を使って署名が行われていることと、その署名プライベート鍵で署名された署名データ自体が必要である。そのため、署名検証公開鍵では、署名プライベート鍵と署名データが関連性保護の対象となる。

保証の必要性

- 有効性の保証とは、公開鍵暗号や署名で使用されるパラメータ、及び公開鍵とそれに対応するプライベート鍵との関係が算術的に正しいことを保証することである。この保証がない場合、公開鍵とプライベート鍵の対応関係が成り立っていないことになる。
- 保有の保証とは、鍵所有者が自らのプライベート鍵を実際に保有していることを保証することである。この保証がない場合、本来の鍵所有者ではない別のエンティティ（不正なエンティティ）が、身元を偽ってプライベート鍵を生成することが可能になる。

[25] CMVP（Cryptographic Module Validation Program）とは、暗号モジュールに搭載されるセキュリティ機能等の確実性・安全性を第三者機関による試験により認証する制度である。攻撃耐性・堅牢さによりレベル 1～4 の 4 段階で認証される。代表的なものとして、米国 FIPS 140 認証、日本 JCMVP 認証がある。

保護期間

鍵が保護されている必要がある期間のことである。なお、機密性が求められる鍵は、保護期間終了後直ちに**破棄されるべきである**。

表 9　鍵の保護要件

鍵タイプ	セキュリティ特性	関連性保護	保証の必要性	保護期間
データ暗号化対称鍵	機密性 完全性 可用性	● 当該鍵を共有しているエンティティ ● 暗号化データ	―	当該鍵の生成から、データのセキュリティ寿命が尽きる又は利用期間が終わるまでのいずれか遅い方の期間
鍵共有対称鍵	機密性 完全性 可用性	● 当該鍵を共有しているエンティティ ● 暗号化された鍵	―	当該鍵の生成から、利用期間が終わる又は暗号化された鍵が保護を必要としなくなるまでのいずれか遅い方の期間
鍵共有プライベート鍵	機密性 完全性 可用性	● 鍵共有公開鍵 ● 暗号化された鍵	保有	当該鍵の生成から交換した全ての鍵の保護期間が終了するまで
鍵共有公開鍵	完全性	● 鍵共有プライベート鍵	有効性	当該鍵の生成から利用期間終了まで
署名プライベート鍵	機密性 完全性	● 署名検証公開鍵	保有	当該鍵の生成から利用期間終了まで
署名検証公開鍵	完全性 可用性	● 署名プライベート鍵 ● 署名データ	有効性	当該鍵の生成から署名データの検証が不要になるまで
認証対称鍵	機密性 完全性 可用性	● 当該鍵を共有しているエンティティ ● 認証データ	―	当該鍵の生成から認証データの検証が不要になるまで
認証プライベート鍵	機密性 完全性	● 認証公開鍵	保有	当該鍵の生成から利用期間終了まで
認証公開鍵	完全性 可用性	● 認証プライベート鍵 ● 認証データ	有効性	当該鍵の生成から認証データの検証が不要になるまで

6.2 鍵の危殆化対策

保護されたデータは、アルゴリズムが安全であったとしても、鍵が危殆化していない場合にのみ安全である。

鍵の危殆化とは、当該鍵の機密性、完全性、又は鍵の所有者との関連付けが喪失した、もしくは喪失したと疑われる状態（4.5 節参照）になることである。例えば、鍵の危殆化が起きると以下のような影響が起こり得る。

- 鍵の機密性が喪失することは、別のエンティティ（不正なエンティティ）が当該鍵を使用して、その鍵を必要とする処理が実行できる可能性があることを意味する。例えば、データ暗号化対称鍵が危殆化した場合、不正なエンティティは、当該鍵を利用して、過去から将来にわたって暗号化された情報を復号（すなわち、情報が正規のエンティティ間の機密ではなくなる）したり、虚偽の情報を暗号化して正規のエンティティに送りつけたりする可能性がある。また、署名プライベート鍵が危殆化した場合は、不正なエンティティが虚偽の情報に署名した可能性が生じるため、当該鍵で署名された全てのデータの完全性と否認防止の特性が疑われる余地が出てくることを意味する。
- 鍵の完全性が喪失することは、鍵が（意図的か偶発的かにかかわらず）変更されたか、別の鍵が置き換えられたかのいずれかであり、当該鍵が正しくない（本来の鍵とは異なるものになっている）ことを意味する。
- 所有者との鍵の関連性が喪失することは、当該鍵を持つエンティティの身元が保証されない（すなわち、当該鍵の所有者が実際に誰であるかわからない）ことを意味する。

鍵の危殆化の可能性や影響を最小限に抑えるために、以下のような危殆化対策を考慮することが望ましい。特に、プライベート鍵及び対称鍵については、所有者に当該鍵の機密性を保護する責任がある。

1. 鍵の利用期間を制限する
2. 一つの鍵で保護されるデータの量を制限する
3. 暗号処理ごとに異なる鍵を使用する
4. 対称鍵やプライベート鍵が平文形式で存在する時間を制限する
5. 平文の対称鍵やプライベート鍵を人間が閲覧できないようにする
6. 平文の対称鍵やプライベート鍵が配置される場所を物理的に保護された“コンテナ”内に制限する
7. 完全性チェックを使用して、鍵の完全性や他のデータとの関連性が危殆化していないことを確認する
8. 鍵が不要になったらすぐに当該鍵を破棄する

7. 運用中における鍵長移行に関する検討の必要性

システムやアプリケーションに必要な暗号処理ごとに、システムやアプリケーションの想定運用終了・廃棄年、鍵長のセキュリティ寿命、及び保護すべき対象データのセキュリティ寿命を考慮し、必要なセキュリティ強度要件を満たす鍵長を選択して**利用すべきである**。

ただ、システムやアプリケーションの運用開始時点での利用環境等によっては、将来的に必要となる高いセキュリティ強度の鍵長を当初から選択すると、対応製品がない、導入コストが許容できないほど高くなる、性能が許容できないほど遅くなるなど、パフォーマンスや導入スケジュール等に悪影響を及ぼす可能性がある。このような場合、例えば、セキュリティ強度を切り替える移行時期を明確化したスケジュールを立案したうえで、システムやアプリケーションの運用寿命の前半に対して適切なセキュリティ強度を有する鍵長を選択して利用するといったことが考えられる。

また、システムやアプリケーションの運用開始時点は想定できなかった（もしくはあえて考慮対象から外した）暗号解読の向上や大規模な量子コンピュータの実現などが現実化し、想定していたよりも早期に使用している鍵長が適切なセキュリティ（暗号学的安全性）を提供できなくなることも起こり得る。

これらのケースでは、システムやアプリケーションの運用寿命の途中で、利用している鍵長のセキュリティ寿命が尽きつつあることを意味するため、そのセキュリティ寿命が尽きる前に、その後に必要となるセキュリティ強度を有する新しい鍵長へ**移行すべきである**。もし、鍵長のセキュリティ寿命が尽き、もはや情報に対して望ましい保護を提供しないと判断された（例えば、“解読された”可能性がある）場合、その鍵長によって保護されている情報は疑わしいと見なされることになる（例えば、当該データの機密性が損なわれていたり、完全性が保証できなくなったりする）。

なお、鍵長の移行だけでは必要なセキュリティ強度が達成できず、利用しているアルゴリズムそのものも同時に移行する必要がある場合には、鍵長だけの移行で必要となるコストや時間よりも多くのコストや時間がかかる可能性があるなど、さらに多くの検討課題が出てくることに留意されたい。

7.1 移行計画策定における論点

新しい鍵長へ移行するのは、システムやアプリケーションの規模や移行対象の鍵長の種類、代替する鍵長の実装状況、データフォーマットやプログラムインタフェースの差異による移行容易性の違いなどにもよるが、多くの場合、非常に時間とコストがかかる作業であることを念頭に置いておく必要がある。実際、過去にあった鍵長における大規模な移行（例：RSA での鍵長 1024 ビットから 2048 ビットへの移行）では、移行準備から移行完了までに 5 年から 10 年単位の時間

がかかっている[26]。

そのため、利用している鍵長のセキュリティ寿命を迎える少なくとも5年前までには、より安全な鍵長への移行計画を**策定すべきである**。その移行計画を立てる際には、いつからどのくらいの期間をかけてどの鍵長に移行するのかを**明確にすべきである**。

以下では、移行のための論点のいくつかを述べる。

7.1.1 通信時及び鍵共有の暗号化における論点

送信側と受信側の両方でより安全な新しい鍵長が実装され利用可能になった時点以降であれば、新しい鍵長だけを使うように切り替えることで移行対策は実現可能である。

なお、移行前に行われた通信や鍵共有について、攻撃者が通信中の暗号化された情報や鍵[27]を収集・保存している可能性を強く想定する必要がある場合、それらの通信内容が解読され、当該情報の機密性が危殆化する可能性がある[28]と**考えるべきである**ことに留意されたい。この場合、別の鍵やアルゴリズムを用いて再暗号化したとしてもセキュリティ上の必要な効果が得られるかどうかは不明である。

このような攻撃に対しては後から防ぐことができないため、こういった攻撃に対する対策が必要であるかどうかについても、移行計画を立てる際に十分に**検討すべきである**。

7.1.2 保管時の暗号化における論点

保管するデータに対して期待されるセキュリティ寿命（当該データの機密性を保持する期間）を考慮に入れることが非常に重要である。

データのセキュリティ寿命全体が鍵長のセキュリティ寿命を超えない範囲にある場合に、当該鍵長のデータ暗号化対称鍵を使って暗号化を**行うべきである**。もしそのような鍵長がサポート（実装）されていないのであれば、より安全な鍵長が実装され利用可能になった後に再暗号化を行うことができるようになるまでは、復号に利用する鍵長のセキュリティ寿命が尽きる期日と同じになるように当該データのセキュリティ寿命を**短縮すべきである**。

保管時の暗号化における移行対策では、新しい鍵長が実装され利用可能となった後の切り替えだけではなく、すでに暗号化された形で保管されているデータについての扱いも検討し、必要な処置を**行うべきである**。

---

[26] 政府機関の情報システムで使用されていたSHA-1及びRSA-1024をSHA-2及びRSA-2048に移行する際には、2008年4月情報セキュリティ政策会議決定を皮切りに、各府省庁に対して2008年度中の移行計画の立案を要請、2009年度に検証システム構築、2010年度から2013年度までのシステム移行期間が設けられた。また、米国でもSHA-1及びRSA-1024をSHA-256及びRSA-2048へ2010年までに移行する方針が表明されたのも2005年である。

[27] 鍵共有が行われた際のセッション鍵が危殆化した場合、当該セッション鍵を利用した暗号通信も同時に危殆化したものと**判断すべきである**。

[28] Store (Harvest)-then-decrypt、Store now & decrypt later、Retrospective Decryptionともいう。このような攻撃は後から防ぐことができないため、システムやアプリケーションの設計・開発時点で必要性について十分に**検討すべきである**。

例えば、すでに暗号化された上で保管されているデータのセキュリティ寿命を延長した場合や何らかの理由で暗号化に利用した鍵長のセキュリティ強度が低下した場合、本来必要とされるセキュリティ強度よりも低い状態で当該データが暗号化されている状態になり得る。その場合は、データの機密性が保たれている間に、より安全な鍵長で当該データの再暗号化をして**保護し直すべきである**。

なお、本来必要とされるセキュリティ強度よりも低い状態であっても、すでに暗号化された形で保管されているデータに対する機密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は、特に移行対策を取ることなく、暗号化されたデータを継続利用（復号）することを考慮してもよい。

7.1.3 署名における論点

署名するデータに対して期待されるセキュリティ寿命（当該データの完全性及び署名者の検証が行える期間）を考慮に入れることが非常に重要である。

署名の署名検証期間全体（すなわち、署名検証用の公開鍵証明書の有効期間）が鍵長のセキュリティ寿命を超えない範囲にある場合に、当該鍵長の署名検証用公開鍵に対応する署名プライベート鍵を使って署名生成を**行うべきである**。もしそのような鍵長がサポート（実装）されていないのであれば、より安全な鍵長が実装され利用可能になるまでは、署名検証に利用する鍵長のセキュリティ寿命が尽きる期日と同じになるように署名検証期間を**短縮すべきである**。

署名における移行対策では、新しい鍵長が実装され利用可能となった後の切り替えだけではなく、有効期間が残っている公開鍵証明書の扱い、及びすでに署名された形で保管されているデータについての扱いも検討し、必要な処置を**行うべきである**。

例えば、すでに署名された形で保管されているデータのセキュリティ寿命を延長した場合や何らかの理由で署名生成に利用した鍵長のセキュリティ強度が低下した場合、本来必要とされるセキュリティ強度よりも低い状態で当該データの署名が生成されている状態になり得る。その場合は、署名の検証が正しく行えている間に、より安全な鍵長を使用した署名を再適用する方法のほか、より安全なアルゴリズムによる再署名、暗号学的タイムスタンプを採用した保存機能や長期署名システムなどを利用するなどの方法により、署名を**保護し直すべきである**。また、関連する公開鍵証明書について、有効期間が残っている場合には、失効処理などの対応が必要となる。

なお、本来必要とされるセキュリティ強度よりも低い状態であっても、すでに署名された形で保管されている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は、特に移行対策を取ることなく、当該データの署名検証を継続することを考慮してもよい。

7.1.4 メッセージ認証における論点

署名の場合と同様に、認証するデータに対して期待されるセキュリティ寿命（当該データの完全性検証が行える期間）を考慮に入れることが非常に重要である。

データの検証期間全体（すなわち、データの完全性を保護する必要がある期間）が鍵長のセキ

ュリティ寿命を超えない範囲にある場合に、当該鍵長の認証対称鍵を使ってメッセージ認証コードの生成を**行うべきである**。もしそのような鍵長がサポート（実装）されていないのであれば、より安全な鍵長が実装され利用可能になるまでは、メッセージ認証コードの検証に利用する鍵長のセキュリティ寿命が尽きる期日と同じになるように検証期間を短縮**すべきである**。

メッセージ認証における移行対策では、新しい鍵長が実装され利用可能となった後の切り替えだけではなく、すでにメッセージ認証コードともに保管されているデータについての扱いも検討し、必要な処置を**行うべきである**。

例えば、すでにメッセージ認証コードとともに保管されているデータのセキュリティ寿命を延長した場合や何らかの理由でメッセージ認証コードの生成に利用した鍵長のセキュリティ強度が低下した場合、本来必要とされるセキュリティ強度よりも低い状態で当該データのメッセージ認証コードが生成されている状態になり得る。その場合は、データの完全性検証が正しく行えている間に、より安全な鍵長で当該データのメッセージ認証コードを再生成して**保護し直すべきである**。

なお、本来必要とされるセキュリティ強度よりも低い状態であっても、すでに生成されたメッセージ認証コードとともに保管されているデータに対する完全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は、特に移行対策を取ることなく、当該データの完全性検証を継続することを考慮してもよい。

7.1.5　エンティティ認証における論点

送信側と受信側の両方でより安全な新しい鍵長が実装され利用可能になった時点以降であれば、新しい鍵長だけを使うように切り替えることで移行対策は実現可能である。

なお、署名によるエンティティ認証の場合で、移行前の公開鍵証明書の有効期間が残っている場合には、失効処理などの**対応が必要となる**。

7.2 システムやアプリケーションの運用寿命の延長に伴う移行にあたっての対応

システムやアプリケーションの運用中の状況の変化により、当該システムやアプリケーションの設計・開発段階で当初想定した運用寿命どおりには運用を終了せず、延長して運用を継続する必要性が生じる場合があり得る。

このような場合、延長の必要性が判明した後、できるだけ早期に、新たに設定される運用寿命をもとに、必要なセキュリティ強度要件を**再評価すべきである**。再評価の結果、

- 求められるセキュリティ強度要件に変化がなく、現在利用中の鍵長でも同じように必要なセキュリティ強度を維持できる場合は、そのまま継続して利用してよい。

- より強力なセキュリティ強度が求められ、現在利用中の鍵長では必要なセキュリティ強度要件を満たすことができない場合には、7.1 節の論点を踏まえ、より安全な鍵長への移行計画をできるだけ早期に策定し、その計画に則って新しい鍵長への移行を**完了すべきである。**

7.3 暗号技術の推定セキュリティ強度の変更に伴う移行にあたっての対応

2.2 節に記載された暗号技術の推定セキュリティ強度の予測の妥当性を確認する観点から、5 年ごと又は必要に応じて、表 3～表 5 の暗号技術の推定セキュリティ強度のレビューを実施し、適宜適切な修正を加えることを計画している。例えば、画期的な暗号解読手法の発明や大規模な量子コンピュータの実現予測時期の精緻化などに伴い、鍵長によってはその推定セキュリティ強度の結果が変更になる可能性がある。

システムやアプリケーションの運用者は、本書が改訂されるタイミングで変更内容を確認し、利用している鍵長についての推定セキュリティ強度が変更されていないかどうかを**確認すべきである。**

利用している鍵長についての推定セキュリティ強度が変更され、当該鍵長では必要なセキュリティ強度要件を満たすことができないことが判明した時は、7.1 節の論点を踏まえ、より安全な鍵長への移行計画をできるだけ早期に策定し、その計画に則って新しい鍵長への移行を**完了すべきである。**

なお、移行に向けた対処方針が別途提示された鍵長を利用している場合には、その対処方針も参考にして移行計画を**検討すべきである。**

7.4 突発的な理由に伴う緊急移行にあたっての対応

可能性は低いものの、あるアルゴリズムに対する極めて画期的な暗号解読手法が発明され、当該アルゴリズムや鍵長の推定セキュリティ強度の急速な低下を引き起こす可能性はゼロではない。そのため、CRYPTREC では、CRYPTREC 暗号リスト掲載のアルゴリズム及び鍵長に対するセキュリティ（暗号学的安全性）を常時監視しており、セキュリティ（暗号学的安全性）が大きく懸念されるような学会発表やニュース報道などに対して、必要に応じて注意喚起情報を発表している。

注意喚起一覧：https://www.cryptrec.go.jp/er.html

利用しているアルゴリズムや鍵長についての注意喚起情報が発表されたとしても、緊急対応を求める旨の記述がなければ、直ちに何らかの対処を求めるというものではない。ただし、内容によっては、その後、CRYPTREC 暗号リスト又は本書での推定セキュリティ強度やセキュリティ

強度要件などの見直しに反映されることがあるので、それらが改訂された際には 7.3 節に従って対処することが望ましい。

なお、可能性は極めて低いものの、全く想定できなかった推定セキュリティ強度の著しい低下により大きな被害の発生が懸念される場合[29]には、緊急対応を求める旨の発表がなされる可能性がある。その際の対処方針によっては、移行対象となったアルゴリズムや鍵長を利用している場合、移行を短期間で終えるための緊急移行計画を検討し、できるだけ早期に**実行すべき**場合もあることに留意されたい。

7.5 量子コンピュータの実現リスクへの対応

現在、大規模な量子コンピュータが実現しても安全な耐量子計算機暗号（PQC: Post-Quantum Cryptography）の標準化選定プロセス[30]を NIST が進めている。また、CRYPTREC 暗号技術評価委員会でもその傘下に暗号技術調査 WG（耐量子計算機暗号）を設置し、PQC の研究動向調査をもとに主要な PQC についてのガイドライン策定を進めている。

今後、これらの活動の進捗状況及び量子コンピュータの進展状況によっては、本書にその成果が取り込まれ、内容が大きく更新される可能性があることに留意されたい。その場合、将来標準化される PQC も代替アルゴリズムの有力な選択肢の一つとなり得る。

その一方、現在の CRYPTREC 暗号リストに掲載されているアルゴリズムの鍵長と PQC の鍵長とでは大きくサイズが異なるため、移行にあたってアプリケーションやインタフェース、データフォーマット、プロトコルなどに大幅な変更が必要となる可能性が高い。その場合、移行のための準備や開発コスト、実際の移行に必要な期間などが従来以上に大きく膨らむ可能性があることに留意されたい。加えて、現在主流の暗号技術とは違い、PQC に特化した暗号解読手法や安全性評価の蓄積、実装脆弱性を回避するための PQC を実装する際のセキュリティ対策（例えば、サイドチャネル攻撃[31]対策）の蓄積といったものが十分に進んでいるとはいえない状況である点も考慮しておく必要がある。

したがって、PQC への移行については、ガイドライン等を参考に、移行の必要性や方法などについても予め十分に検討し、移行計画を慎重に策定したうえで**実施するのが望ましい**。利用環境によっては、PQC への完全な移行ではなく、PQC と現在主流の暗号技術との併用を視野に入れることも考えられる。

---

29 ちなみに、2000 年の CRYPTREC 発足以来、今までにそのようなケースが発生したことは一度もない。

30 NIST Post-Quantum Cryptography Standardization, https://csrc.nist.gov/Projects/post-quantum-cryptography/post-quantum-cryptography-standardization

31 暗号技術が実装された暗号モジュールやプログラム、チップなどから、実際に暗号保護を行う際に漏えいする物理的情報（消費電力、処理時間、電磁波など）を測定することによって、内部の動作状況を推定し、暗号鍵などの秘密情報を入手する攻撃手法のこと。電力解析攻撃、タイミング攻撃などが有名。アルゴリズムではなく実装物への攻撃なので、アルゴリズムそのものは安全であったとしても、実装された暗号モジュールやプログラム、チップが脆弱であったために暗号解読されたというケースは多い。

Appendix　参考情報

[1] Cryptographic Mechanisms: Recommendations and Key Lengths, TR-02102-1 v2020-01, BSI, 03/2020.
https://www.bsi.bund.de/SharedDocs/Downloads/EN/BSI/Publications/TechGuidelines/TG02102/BSI-TR-02102-1.html

表 10　BSI（独）のセキュリティ強度選択基準（1.1 節、1.2 節）

2020～2022	（要件）100 ビット以上のセキュリティ強度であること （推奨）共通鍵暗号：128 ビットセキュリティ メッセージ認証コード：128 ビットセキュリティ 楕円曲線以外の公開鍵暗号（RSA, DH など）：100 ビットセキュリティ（鍵長 2000 ビット） 楕円曲線の公開鍵暗号（ECDSA など）：120 ビットセキュリティ（鍵長 250 ビット）
2023～2026	（要件）　120 ビット以上のセキュリティ強度であること （推奨）共通鍵暗号：128 ビットセキュリティ メッセージ認証コード：128 ビットセキュリティ 楕円曲線以外の公開鍵暗号（RSA, DH など）：120 ビットセキュリティ（鍵長 3000 ビット） 楕円曲線の公開鍵暗号（ECDSA など）：120 ビットセキュリティ（鍵長 250 ビット）

[2] Recommendation for Key Management, Special Publication 800-57 Part 1 Rev. 5, NIST, 05/2020.
https://csrc.nist.gov/publications/detail/sp/800-57-part-1/rev-5/final

表 11　NIST（米）のセキュリティ強度選択基準（5.6.3 節）

2020～2030	（要件）新規データの保護（暗号化、署名生成など）は 112 ビット以上のセキュリティ強度であること。但し、2024 年以降は、3-key Triple DES は利用不可 保護済データの処理（復号、署名検証など）は 2-key Triple DES、1024 ビット RSA、SHA-1 相当以上のセキュリティ強度であること
2031～	（要件）新規データの保護（暗号化、署名生成など）は 128 ビット以上のセキュリティ強度であること 保護済データの処理（復号、署名検証など）は 2-key Triple DES、1024 ビット RSA、SHA-1 相当以上のセキュリティ強度であること

[3] Mécanismes cryptographiques - Règles et recommandations, Rev. 2.03, ANSSI, 02/2014
https://www.ssi.gouv.fr/uploads/2014/11/RGS_v-2-0_B1.pdf

表 12　ANSSI（仏）のセキュリティ強度選択基準（2.1 節、2.2 節、2.3 節）

2014～2030	（要件）共通鍵暗号：128 ビット以上のセキュリティ強度。なお、ブロック暗号のブロック長は 128 ビット 楕円曲線以外の公開鍵暗号（RSA, DH など）：112 ビット以上のセキュリティ強度（鍵長 2048 ビット以上） 楕円曲線の公開鍵暗号（ECDSA など）：128 ビット以上のセキュリティ強度（鍵長 256 ビット以上） ハッシュ関数：128 ビット以上のセキュリティ強度（ハッシュ長 256 ビット以上） （推奨）楕円曲線以外の公開鍵暗号（RSA, DH など）：128 ビット以上のセキュリティ強度（鍵長 3072 ビット以上）
2031～	（要件）共通鍵暗号：128 ビット以上のセキュリティ強度。なお、ブロック暗号のブロック長は 128 ビット 楕円曲線以外の公開鍵暗号（RSA, DH など）：128 ビット以上のセキュリティ強度（鍵長 3072 ビット以上） 楕円曲線の公開鍵暗号（ECDSA など）：128 ビット以上のセキュリティ強度（鍵長 256 ビット以上） ハッシュ関数：128 ビット以上のセキュリティ強度（ハッシュ長 256 ビット以上）

* ビットセキュリティ自体の表示はなし。鍵長・ハッシュ長からの推定

[4] Commercial National Security Algorithm, National Security Agency (NSA), 01/2016.
https://apps.nsa.gov/iaarchive/programs/iad-initiatives/cnsa-suite.cfm

表 13　NSA（米）のセキュリティ強度選択基準

TOP SECRET までの保護	（要件）共通鍵暗号：256 ビットセキュリティ 楕円曲線以外の公開鍵暗号（RSA, DH など）：128 ビット以上のセキュリティ強度（鍵長 3072 ビット以上） 楕円曲線の公開鍵暗号（ECDSA など）：192 ビットセキュリティ（鍵長 384 ビット） ハッシュ関数：192 ビットセキュリティ（ハッシュ長 384 ビット）

[5] Algorithms, Key Size and Protocols Report (2018), H2020-ICT-2014 – Project 645421, D5.4, ECRYPT-CSA, 02/2018.
https://www.ecrypt.eu.org/csa/documents/D5.4-FinalAlgKeySizeProt.pdf

表 14　ECRYPT（欧州）のセキュリティ強度選択基準（4.6 節）

互換性維持	
2018 ～ 2028 (near term use) 短期の利用	（要件）共通鍵暗号：128 ビットセキュリティ 楕円曲線以外の公開鍵暗号（RSA, DH など）：128 ビットセキュリティ（鍵長 3072 ビット） 楕円曲線の公開鍵暗号（ECDSA など）：128 ビットセキュリティ（鍵長 256 ビット） ハッシュ関数：128 ビットセキュリティ（ハッシュ長 256 ビット）
2018 ～ 2068 (long term use) 長期の利用	（要件）共通鍵暗号：256 ビットセキュリティ 楕円曲線以外の公開鍵暗号（RSA, DH など）：256 ビットセキュリティ（鍵長 15360 ビット） 楕円曲線の公開鍵暗号（ECDSA など）：256 ビットセキュリティ（鍵長 512 ビット） ハッシュ関数：256 ビットセキュリティ（ハッシュ長 512 ビット）

[6] Selecting Cryptographic Key Sizes, Arjen K. Lenstra and Eric R. Verheul, Journal of Cryptology, vol. 14, p. 255-293, 2001.
https://link.springer.com/content/pdf/10.1007/s00145-001-0009-4.pdf
[7] Key Lengths, Arjen K. Lenstra, The Handbook of Information Security, 06/2004.
http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.694.8206

表 15　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 年程度なら完全解読が困難と期待される）ビットセキュリティ
([3] Figure 6、[6] Table 1、[7] 2 節式(2))

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

[8] CRYPTREC Report 2021

https://www.cryptrec.go.jp/report/cryptrec-rp-2000-2021.pdf

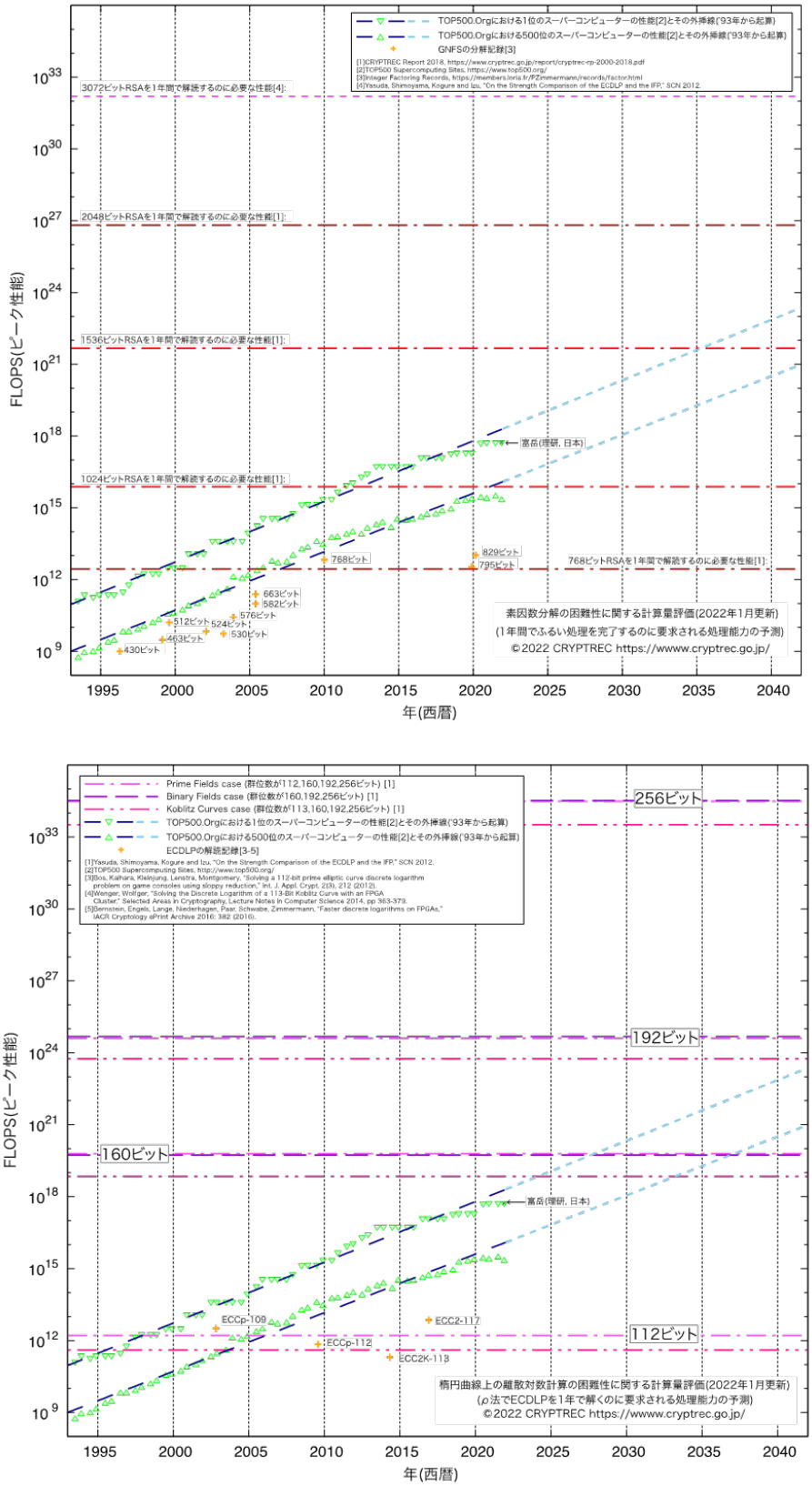

図 7　1 年で解読するのに必要な性能が達成できると見込まれる時期（図 3.2-1、図 3.2-2）

発行日　2022 年 7 月 1 日　　第 1.0 版

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　セキュリティ技術評価部　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