

NIST Special Publication 800-57 Part 3
Revision 1

鍵管理における推奨事項

**第三部：アプリケーション特有の
鍵管理ガイダンス**

Elaine Barker
Quynh Dang

This Publication is available free of Charge from:
<http://dx.doi.org/10.6028/NIST.SP.800-57pt3r1>

コンピュータ セキュリティ

NIST
**National Institute of
Standards and Technology**
U.S. Department of Commerce

この文書は以下の団体によって翻訳監修されています

IPA 独立行政法人 情報処理推進機構
INFORMATION-TECHNOLOGY PROMOTION AGENCY, JAPAN

NIST Special Publication 800-57 Part 3
Revision 1

鍵管理における推奨事項

**第三部：アプリケーション特有の
鍵管理ガイダンス**

Elaine Barker
Quynh Dang
コンピュータセキュリティ部門
情報技術研究所

This Publication is available free of Charge from:
<http://dx.doi.org/10.6028/NIST.SP.800-57pt3r1>

2015年1月



米国商務省
Penny Pritzker 長官

米国国立標準技術研究所
Willie May 標準技術担当次官兼所長代行

発行機関

本文書は、米国国立標準技術研究所（NIST：National Institute of Standards and Technology、以下、NIST と称す）によって、連邦情報セキュリティマネジメント法（FISMA：Federal Information Security Management Act）公法（P.L.）107-347 に基づく法的責任を推進するために開発された。NIST は、連邦情報システムの最小限の要求事項を含め情報セキュリティ標準およびガイドラインを開発する責務があるが、このような標準およびガイドラインは国家安全保障に適用されてはならず、このようなシステムについての政策的権限を有する適切な連邦機関の明確な承認が必要となる。このガイドラインは、行政管理予算局（OMB：Office of Management and Budget）による通達（Circular）A-130、第 8b（3）項、*政府機関の情報システムの保護（Securing Agency Information Systems）* の要求事項に一致しており、これは通達 A-130、附属書Ⅳ：重要部門の分析で分析されているとおりである。補足情報は通達 A-130、附属書Ⅲ、*連邦自動化情報資源* で提供されている。

本文書における一切は、商務長官が法的権威に基づき連邦政府に対して義務および拘束力を与えた標準およびガイドラインを否定するものではない。また、これらのガイドラインは、商務長官、行政管理予算局長、または他のすべての連邦政府当局者の既存の権威に変更を加えたり、これらに取って代わるものと解釈したりしてはならない。本文書は、非政府組織が自由意思で使用することもでき、米国における著作権の制約はないが、NIST に帰属する。

National Institute of Standards and Technology Special Publication 800-57 Part 3,
Revision 1

Natl. Inst. Stand. Technol. Spec. Publ. 800-57 Part 3, Revision 1, 102 pages (January 2015)
CODEN: NSPUE2

この公表文書は、以下から無料で利用可能である：

<http://dx.doi.org/10.6028/NIST.SP.800-57pt3r1>

本文書中で特定される商業的組織、装置、資料は、実験手順または概念を適切に説明するためのものである。このような特定は、NIST による推奨または同意を意味するものではなく、これらの組織、資料、または装置が、その目的のために利用可能な最善のものであることを意味している訳ではない。

与えられた法的責任に従い、NIST によって現在開発中のその他の文書への参照が本文書にあるかもしれない。本文書におけるその情報は、概念および方法論を含め、このような関連文書の完成前であっても連邦政府によって利用されるかもしれない。したがって、それぞれの文書が完成されるまで、現在の要求事項、ガイドライン、および手順は存在する限り運用の効力を有する。計画および移行目的に関して、連邦政府は、NIST によるこれらの新しい文書の開発に密接に従うことを希望するかもしれない。

公開コメント期間中に組織がすべてのドラフト文書をレビューし、NIST へフィードバックを提供するよう奨励する。上記以外のすべての NIST コンピュータセキュリティ部門の文書は、<http://csrc.nist.gov/publications> において利用可能である。

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: SP80057Part3@nist.gov

コンピュータシステムの技術に関する報告書

米国国立標準技術研究所（NIST : National Institute of Standards and Technology、以下、NIST と称す）の情報技術ラボラトリ（ITL : Information Technology Laboratory、以下、ITL と称す）は、国家の計測および標準に関する基盤において技術的リーダーシップを提供することにより、米国の経済および社会福祉に貢献している。ITL は、テストの開発、テスト技法の開発、参照データの作成、概念実証の実施および技術的分析を通じて、情報技術の開発と生産的利用の発展に努めている。ITL の責務は、連邦政府の情報システムにおいて、国家安全保障に関連する情報以外の情報に対する費用対効果の高いセキュリティとプライバシーを実現するための、技術面、物理面、管理面及び運用面での標準およびガイドラインを策定することが含まれる。本 Special Publication 800 シリーズは、情報システムセキュリティに関する ITL の調査、ガイドラインおよび公共福祉のために教育や援助を行う努力、ならびに産業界、政府機関および学術機関との共同活動について報告する。

要旨

NIST Special Publication 800-57 は、暗号鍵管理に関するガイダンスを提供する。本文書は 3 部から構成される。第一部は、暗号鍵材料の管理に関する一般的なガイダンスおよびベストプラクティスを提供する。第二部は、米国政府機関向けに方針およびセキュリティ計画の要求事項に関するガイダンスを提供する。最後に、第三部は、現時点でのシステムの暗号機能を使用する際のガイダンスを提供する。

キーワード

認定； 保証； 認証； 権限付与； 可用性； バックアップ； 証明； 危殆化； 機密性； 暗号解析； 暗号鍵； 暗号モジュール； デジタル署名； 鍵管理； 鍵管理方針； 鍵回復； プライベート鍵； 公開鍵； 公開鍵基盤； セキュリティ計画； トラストアンカー； 検証。

謝辞

SP 800-57、第三部の本バージョンの共著者たちは、本文書の以前の共著者、William Burr 氏、Alicia Jones 氏、Timothy Polk 氏、Scott Rose 氏および Miles Smid 氏の貢献に深く感謝の意を表します。我々は、また Katrin Reitsma 氏、Sheila Frankel 氏、David Cooper 氏、Judith Spencer 氏、国家安全保障局 (NSA) および本発行の文書の品質および有用性の改善に寄与したその他多くの公共および民間の方々の思慮深く建設的なコメントによる貢献に対しても感謝の意を表します。

本文書は、原典に沿ってできるだけ忠実に翻訳するよう努めていますが、完全性、正確性を保証するものではありません。

翻訳監修主体は、本文書に記載されている情報より生じる損失または損害に対して、いかなる人物あるいは団体についても責任を負うものではありません。

目次

1	序説	1
1.1	目的	2
1.2	要求事項についての用語	3
1.3	汎用プロトコルの検討	3
1.3.1	実装が必須 対 実装がオプション	3
1.3.2	暗号的なネゴシエーション	4
1.3.3	1つまたは複数の用途の鍵	5
1.3.4	アルゴリズムと鍵サイズの移行	5
2	公開鍵基盤 (PKI)	7
2.1	説明	7
2.2	セキュリティおよび適合性の問題	9
2.2.1	推奨される鍵サイズとアルゴリズム	9
2.3	調達ガイダンス	11
2.3.1	CA/RA ソフトウェアおよびハードウェア	11
2.3.2	OCSP レスポンダ	12
2.3.3	暗号モジュール	13
2.3.4	鍵回復サーバ	13
2.3.5	依拠当事者のソフトウェア	13
2.3.6	クライアントソフトウェア	14
2.4	システムの設置者/管理者のための推奨事項	14
2.4.1	証明書発行	14
2.4.2	証明書失効要求	15
2.4.3	証明書失効リストの生成	16
2.4.4	証明書と CRL の配付のための PKI リポジトリ	16
2.4.5	OCSP レスポンダ	16
2.4.6	バックアップとアーカイブ	17
2.4.7	依拠当事者の統合と設定	17
2.5	利用者ガイダンス (加入者)	17
3	インターネットプロトコルセキュリティ (IPsec)	19
3.1	説明	19
3.2	セキュリティおよび適合性の問題	20
3.2.1	暗号アルゴリズム	20
3.2.2	追加の推奨事項	24

3.3	調達ガイダンス.....	24
3.4	システムインストーラのための推奨事項.....	24
3.5	システム管理者のための推奨事項.....	24
3.6	エンドユーザのための推奨事項.....	24
4	トランスポート層セキュリティ (TLS)	25
5	セキュア/多目的インターネットメール拡張 (S/MIME)	26
5.1	説明.....	26
5.2	セキュリティおよび適合性の問題.....	26
5.3	調達ガイダンス.....	28
5.4	システムインストーラのための推奨事項.....	28
5.5	システム管理者のための推奨事項.....	29
5.6	エンドユーザのための推奨事項.....	29
6	ケルベロス (Kerberos)	30
6.1	説明.....	30
6.2	セキュリティおよび適合性の問題.....	32
6.3	調達ガイドライン.....	33
6.4	システムインストーラのための推奨事項.....	34
6.5	システム管理者のための推奨事項.....	35
6.6	エンドユーザのための推奨事項.....	35
7	無線回線経由の鍵更新 (OTAR) 鍵管理メッセージ (KMM)	36
7.1	説明.....	36
7.2	セキュリティおよび適合性の問題.....	37
7.2.1	暗号アルゴリズム.....	37
7.2.2	メッセージ認証と暗号周期.....	37
7.2.3	鍵の用途.....	37
7.2.4	バックアップ.....	37
7.2.5	鍵更新.....	38
7.2.6	乱数ビット生成器.....	38
7.3	調達ガイダンス.....	38
7.4	システムインストーラのための推奨事項.....	38
7.5	システム管理者のための推奨事項.....	39
7.6	エンドユーザのための推奨事項.....	39
8	ドメインネームシステム セキュリティ拡張 (DNSSEC)	40

8.1	説明	40
8.1.1	DNS データ 認証	41
8.1.2	DNS トランザクション 認証	41
8.1.3	DNS 暗号アルゴリズム/スキーム、モードおよび組合せ	42
8.1.4	鍵サイズについての特別な検討	43
8.1.5	NSEC3 のための特別な検討	44
8.2	セキュリティ/適合性の問題	44
8.3	調達ガイダンス	45
8.4	システムインストーラのための推奨事項	45
8.4.1	システムインストーラのための推奨事項 (権威サーバ)	45
8.4.2	システムインストーラのための推奨事項 (キャッシュ再帰サーバ)	45
8.4.3	システムインストーラのための推奨事項 (クライアントシステム)	46
8.5	システム管理者のための推奨事項	46
8.5.1	システム管理者のための推奨事項 (権威サーバ)	46
8.5.2	システム管理者のための推奨事項 (キャッシュ再帰サーバ)	46
8.5.3	システム管理者のための推奨事項 (クライアントシステム)	46
8.6	エンドユーザのための推奨事項	47
9	暗号化ファイルシステム (EFS)	48
9.1	説明	48
9.1.1	必要とされる鍵の数	48
9.1.2	ファイル暗号化で使用される対称鍵へのアクセス	50
9.2	セキュリティおよび適合性の問題	51
9.3	調達担当官のための推奨事項	52
9.4	システムインストーラのための推奨事項	52
9.5	システム管理者のための推奨事項	52
9.6	エンドユーザのための推奨事項	53
10	セキュアシェル (SSH)	54
10.1	説明	54
10.1.1	トランスポート層プロトコル (SSH-TLP)	54
10.1.2	利用者認証プロトコル (UAP)	54
10.1.3	コネクションプロトコル (CP)	55
10.2	セキュリティおよび適合性の問題	55
10.2.1	TLP 問題	55
10.2.2	UAP 問題	58
10.3	調達ガイダンス	59
10.4	システムインストーラのための推奨事項	60

10.5 システム管理者のための推奨事項.....	60
10.6 エンドユーザのための推奨事項	61
附属書 A : 用語	62
附属書 B : 略語	68
附属書 C : 初心者エンドユーザへの言葉	70
附属書 D : 参考文献.....	71
附属書 E : 改訂履歴	83

鍵管理における推奨事項

第三部：アプリケーション特有の鍵管理ガイダンス

1 序説

鍵管理における推奨事項の第三部、アプリケーション特有の鍵管理ガイダンスは、システム管理者およびシステムのインストーラが製品の可用性と組織のニーズに基づいてアプリケーションを適切に確保することを助けること、また、将来の調達に関する組織的な意思決定を支援することが、主に意図されている。また、本文書は、アプリケーションの通常利用の制御から外れたアプリケーションオプションに関する情報をエンドユーザに提供する。推奨事項は一連の選択されたアプリケーションらに与えられる、即ち：

[セクション 2](#) – 公開鍵基盤 (PKI)

[セクション 3](#) – インターネットプロトコルセキュリティ (IPsec)

[セクション 4](#) – トランスポート層セキュリティ (TLS)

[セクション 5](#) – セキュア/多目的インターネットメール拡張 (S/MIME)

[セクション 6](#) – ケルベロス (Kerberos)

[セクション 7](#) – デジタル無線の無線回線経由の鍵更新 (OTAR)

[セクション 8](#) – ドメインネームシステム セキュリティ拡張 (DNSSEC)

[セクション 9](#) – 暗号化ファイルシステム (EFS)

[セクション 10](#) – セキュアシェル (SSH)

各トピックとして以下が提供される：

- ・ セキュリティガイダンスの背景を提供することを意図した検討中のシステムの簡潔な説明、
- ・ 推奨アルゴリズムスイートと鍵サイズおよび関連するセキュリティおよび適合性の問題、
- ・ 連邦政府情報の保護のための現在の形のメカニズムの使用に関する推奨事項、
- ・ 鍵管理処理のセキュリティの有効性に影響する可能性のあるセキュリティ上の考慮事項、
- ・ 調達決定者、システムインストーラ、システム管理者およびエンドユーザへの一般推奨事項。

[セクション 10](#)に続き、5つの附属書として、[用語](#)、[略語](#)の説明、鍵の取得と使用に関する[初心者やエンドユーザのための基本的な情報](#)、本書に記載された文書についての[参考文献](#)、および[本改訂版へ適用された変更](#)がある。

本書は、現在の製品と技術的な仕様の包括的な見解を反映していない。本書の将来のバージョンでは、扱われているトピックへのアップデートが含まれる、また、新しい技術が広く実装されて追加された内容を含むかもしれない。

1.1 目的

鍵管理における推奨事項の第三部、アプリケーション特有の鍵管理ガイダンスが主として意図するのは現在利用可能な暗号メカニズムに関連する鍵管理問題に対処することである。鍵管理における推奨事項の第一部、一般事項ガイダンスは、様々なクラスの暗号鍵材料の生成と使用に関する“ベストプラクティス”に関する利用者、開発者、システム管理者のための基本的な鍵管理ガイダンスを含んでいる[[SP 800-57 Part 1](#)]。推奨事項の第二部、組織と管理の一般要求事項は、組織内での暗号鍵管理の確立を支援するためのフレームワークと一般事項ガイダンス、および連邦政府機関のため法的観点からの鍵管理と方針に基づくセキュリティ計画の要求事項を満たすための原則を提供する。

本書、推奨事項の第三部は、現在利用可能な技術を用いて新しいシステムの調達について決定を行う人々¹のためと同様に、既存の鍵管理基盤、プロトコルおよびその他アプリケーションのシステムインストーラ、システム管理者およびエンドユーザのために設計されている。自らのシステムのインストーラ、システム管理者、または調達機関としての役割を果たすエンドユーザは、管理者、インストーラ、および調達者ために意図された本ガイダンスが有益であることにおそらく気付くだろう。集中管理された組織においては、組織の管理者はすべてのエンドユーザのガイダンスの基盤として機能するセキュリティ方針を確立しなければならない。

推奨事項は、保管データと通信データを保護するよう設計されたメカニズムのために作成されている。本書は、既存の標準または実装指令の完全な言い換えを提供していない。この詳細レベルの標準とガイドラインは適切な箇所が参照されている。

第三部で対応する、それぞれの鍵管理基盤、プロトコル、およびアプリケーションについて、以下が提供される：

- セキュリティガイダンスの背景を提供することを意図した検討中のシステムの簡潔な説明、
- 推奨アルゴリズムスイートと鍵サイズおよび関連するセキュリティおよび適合性の問題、
- 連邦政府情報の保護のための現在の形のメカニズムの使用に関する推奨事項、
- 鍵管理処理のセキュリティの有効性に影響する可能性のあるセキュリティ上の考慮事項、
- 調達決定者、システムインストーラ、システム管理者およびエンドユーザへの一般推奨事項。

所与のアプリケーションまたはシステム内での鍵または鍵ペアを取得、保管または配送すべき方法のロジスティクスはアプリケーションおよび実装依存であり、本書の適用範囲外である。大規模な連邦政府システムにおいて、これらの機能は、システム管理者によってしばしば取り扱われ、またはシステム管理者からの直接のガイダンスで完了される。これらの任務に自身で対応するエンドユーザのために、情報提供するための附属書がエンドユーザを正しい方向に導くための一般的な情報と共に含まれている。

本推奨事項で対応される何らかの基盤、プロトコルおよびアプリケーションは時間を掛けて詳細化され、または置き換えられるので、ここで提供されるガイダンスは廃止となるだろう。同様に、新しい基盤、プロトコル、およびアプリケーションが開発されることが予想される。本書はメカニズムと手法が発達するにつれアップデートされるが、常に現在の製品と技術仕様の包括的な観点を反映しているとは限らない。したがって、ある基盤、プロトコル、またはアプリケーションの具体的なバージョンへのガイダンスの特定要素の実装されたメカニズムへの適用可能性の評価を可能にするために、バージョン番号またはその他の実装ステータス情報への参照が提供される。

第三部に記述された多くのアプリケーションは、米国政府機関によって現在利用されていることに注意すること。これらのアプリケーションのいくつかは、本推奨事項の第一部のリリースの前に開発され、

¹ 必ずしも調達者であるとは限らないが、使用される IT 製品の決定を行うような人である。

実装されたものであり、従って第一部 [\[SP 800-57 Part 1\]](#) で特定された原則のすべてに従っていないかもしれない。これらのアプリケーションの現在の実装の使用は、より注意深く設計されたアプリケーションが利用可能となるまで必要であるかもしれない。NIST 標準とガイドラインに適合しないようなそれぞれの実装が関連するリスクについて評価されること、および本推奨事項で検討されるようなリスクを低減するために取られるステップは、非常に重要である。

1.2 要求事項についての用語

本推奨事項は、しばしば、“要求事項”としての用語を使用する；これらの用語は、以下のような意味を本書において持つ：

1. **しなければならない (shall)** :この用語は、連邦政府情報処理標準 (FIPS) の要求事項または本推奨事項に適用主張するために満たさなければならない要求事項を示すために使用される。**しなければならない (shall)** の反対語は、**してはならない (shall not)** であることに注意すること。
2. **すべきである (should)** :この用語は、重要な推奨事項を示すために使用される。推奨事項を無視することは、望ましくない結果を招く可能性がある。保護されたメッセージの受容が通常使用されるようにするための推奨事項を無視すると、承認されていない暗号が相互運用上の問題を発生させるかもしれない。承認されたほとんど使用されていない暗号メカニズムを持つ新しい製品を選択するために推奨事項を無視することは、連邦政府システムの保護に直ちに不適切となるようなメカニズムから遠ざけるには、組織にとってお粗末であるかもしれない。**すべきである (should)** の反対語は、**すべきではない (should not)** であることに注意すること。

1.3 汎用プロトコルの検討

第三部において議論されるプロトコルに関連する一般的な課題が数多くある。

これらの課題の4つが本文書の全体を通して繰り返される概念を読者に慣れ親しんでいただき、また今後の議論の組み立てを支援するため、簡潔に議論されている：

- ・ 実装が必須 対 実装がオプション、
- ・ 暗号ネゴシエーション、
- ・ 単用途または多用途の鍵、および
- ・ アルゴリズムと鍵サイズの移行

1.3.1 実装が必須 対 実装がオプション

本文書において記述された多くの暗号セキュリティサービスは、公的な標準(例. **Internet Engineering Task Force (IETF) Request for Comment (RFC)**、**American National Standards**、他)に基づいている。これらの標準において、アルゴリズムは実装が必須、または実装がオプションと頻繁に記述される。これらの用語のどちらも、アルゴリズムのセキュリティについての情報を提供しない。

製品間の相互運用性を許容するような、公的な標準を満たすあらゆる製品において、実装が必須のアルゴリズムがあるだろう。

実装がオプションのアルゴリズムは、将来のバージョンの標準において、実装が必須となるような次世代のアルゴリズムである傾向がある。ハードウェアまたはソフトウェアアップデートの必要性から相互運用性の問題までの範囲のさまざまな理由により、これらの新しいアルゴリズムの幅広い使用に遅れが出るのが考えられる。たとえば、**S/MIME** プロトコルでの実装がオプションであるようなアルゴリズムは、システムの暗号モジュールによって現在サポートされていない。

上記の定義のとおり、用語“**しなければならない (Shall)**”と“**すべきである (Should)**”は、アルゴリズムが連邦政府コンピュータネットワーク上での使用に適切なセキュリティを有するかどうかにつ

いての情報を提供するために使用される。このように、それらは、適切なセキュリティを提供しないような実装が必須のアルゴリズム（例、データ暗号化標準（DES）または RC2）があるかもしれない、また本文書はそれらが使用されてはならない（**shall not**）というであろう。同様に、より大きなセキュリティを有する実装がオプションであるアルゴリズム（例、高度暗号化標準（AES）等）があるかもしれない、また本文書はそれらのアルゴリズムが所与の状況において使用されるべきである（**Should**）、または使用されなければならない（**Shall**）というであろう。

異なるシステム上の 2 人の利用者が通信したいとき、または同じシステム上で動作する異なるアプリケーションのために異なるセキュリティレベルが要求されるかもしれないときに、実装が必須、および実装がオプションの間の違いが重要となる。次のセクションにて暗号的なネゴシエーションについてさらに議論される。

1.3.2 暗号的なネゴシエーション

本推奨事項の第一部および第二部では、適切な暗号アルゴリズムの選択および対応する暗号鍵の管理についての健全な基盤を確立している。しかし、これらのガイドラインの実施は、特定のアルゴリズムや鍵サイズが利用できないこと、通信相手による選択またはその他のシステムの制限を含めて、多くの理由から問題のある可能性がある。サーバが利用可能なアルゴリズムを決定するとき、サーバは、最高レベルのセキュリティを提供するものよりもむしろ、システム全体のパフォーマンスを最適化するようなアルゴリズムを選択するかもしれない。

同じ目的のために複数のアルゴリズムをサポートされるような、いくつかの多者間のプロトコルにおいて、クライアントは、そのプロトコル内でのネゴシエーションを通して、第一部および第二部における規則を実施することができる。いくつかのプロトコル（例、S/MIME）は、受信側クライアントとネゴシエーションすることなしに、送信側クライアントが暗号アルゴリズムを選択することを許容している。例えば、受信側クライアントは DES を用いて暗号化され、512 ビット RSA 鍵²で署名された署名付きの暗号化 S/MIME 電子メールメッセージを受信するかもしれない。このようなメッセージの拒否は、必ずしもセキュリティを拡張しない（この場合、メッセージはすでにインターネット経由で送信済である）、しかし受信側の利用者は、デジタル署名および内容の暗号化によって偽って提供されるセキュリティサービスが疑われ、当てにならないことをよくわきまえているべきである。メッセージを拒否するかプロトコルを終了することが適切かもしれない。リスク評定と次の組織方針が適切な行動指針を決定するために要求されるかもしれない。

その他のプロトコル（例、トランスポート層セキュリティ（TLS））において、クライアントは、いくつかの選択肢を提案する、そしてサーバは、プロトコルのネゴシエーションフェーズの間に提案されたリストから選択する。ネゴシエーションがサポートされる場合、プロトコルは、暗号スイートをネゴシエートするか、またはそれぞれのアルゴリズムを独立にネゴシエートするように設計されるかもしれない。いずれの場合においても、クライアントまたはサーバは、クライアントまたはサーバの望ましいアルゴリズムおよび多者の提案したアルゴリズムが同じセキュリティ強度でないような、または承認されたアルゴリズムが利用可能でないような状況に直面するかもしれない。

別の問題として、プロトコルがアルゴリズムをネゴシエートするが、鍵サイズについてはネゴシエートしないように設計されているときが挙げられる。このような場合では、クライアントは承認されたアルゴリズムだが、不適切な鍵サイズで通信することがわかる。例えば、RSA 署名についてネゴシエーションした後、クライアントは 512 ビット RSA 鍵³で署名したメッセージを得るかもしれない。

推奨事項の第一部と第二部を実施することは、システムまたはアプリケーションの設計決定によって複雑にもなるかもしれない。システムは暗号アルゴリズムのためのアプリケーション特有の管理策を持つ

² DES アルゴリズムと 512 ビット RSA 鍵は適切なセキュリティを提供しない（[\[SP 800-57 Part 1\]](#)を参照）。

³ 512 ビット RSA 鍵は受け入れ可能なセキュリティ強度を提供しない（[\[SP 800-57 Part 1\]](#)を参照）。

ているかもしれず、またはシステム全体の管理策を持っているかもしれない。例えば、システム設計が TDEA の使用のみを許容するかもしれないが、利用者は、あるアプリケーションに対して AES 使用に限定し、また別のアプリケーションに対して TDEA 使用に限定したいと望むかもしれない。しばしば公開鍵サイズの制限のみがルート認証局 (CA) 鍵の選択を通して間接的な制限となる ([セクション 2.1](#) を参照)。

所与の通信プロトコルで利用可能なアルゴリズムや鍵サイズに様々なものがあるとき、以下の質問が対処するために必要となる：

- ・ ネゴシエーションは、必須か、オプションか、またはサポートされないか？
- ・ ネゴシエーションがサポートされる時、誰が使用される暗号メカニズムを提案するか、および選択基準は何か？
- ・ ネゴシエーションが事前定義の暗号スイートに基づくか、またはそれぞれのアルゴリズムは独立して提案されるか？
- ・ ネゴシエーションの粒度は何か？単にアルゴリズム、アルゴリズムと鍵サイズ、アルゴリズムおよび/または鍵サイズ、またはプロトコルバージョン？
- ・ 何が規定可能でないか？

複数のアルゴリズム設定における通信セキュリティの保証における良い出発点は、以下のとおりである：

- ・ アルゴリズムのリストをシステムの利用者に適した最良のもの、および相互運用性に必要とされるものへの適用に制限する、
- ・ 不適切なレベルの保護を用いる送信メッセージを許容しないような方針を適用する、
- ・ 適切な保護なしに受信したメッセージへの応答方法を説明する方針を適用する、および
- ・ 承認されていないアルゴリズムまたは不適切な鍵サイズを用いる相手とセキュアな通信の必要性に直面したとき、何をすべきかについて説明するような方針を適用する。

1.3.3 1つまたは複数の用途の鍵

本推奨事項の第一部からの主要な要点は、一般に、鍵は複数の暗号目的で使用してはならないということである。例えば、同じ鍵は、デジタル署名を生成するためおよびその他の鍵材料を確立するために使用されてはならない ([\[SP 800-57 Part 1\]](#)、セクション 8.1.5.1.1.2) を本ガイダンスに対する稀な例外について、参照)。デジタル署名用鍵は、例えば、特定のアプリケーション (例. 電子メールへの署名) のみのために使用可能であるか、または複数のアプリケーション (例. 電子メールへの署名と文書の署名の両方) のために使用が可能であるか、判りづらい。いくつかの場合、他のアプリケーションと鍵を共有することは、アプリケーションにとって受け入れ可能であるかもしれない。その他の場合、鍵を共有することは、望ましくないかもしれない。例えば、ベストプラクティスはサーバの TLS 鍵がその他のアプリケーションをサポートするために使用されるべきでないことを示している。たとえ、鍵が同じ暗号操作 (例. デジタル署名) を実行するために使用されるとしても、鍵を共有することは不適切であるかもしれない、なぜなら 1つのアプリケーションが 1つのサービス (例. 認証) を提供しているときに、2番目のアプリケーションが異なるサービス (例. 否認防止) を提供している可能性があるためである。複数のアプリケーションのために鍵を使用することは、悪い考えかもしれないことを思い出すことは重要である。

政府機関は、複数のアプリケーションに同じ鍵の使用を検討するとき、リスク評価を実行するべきである。

1.3.4 アルゴリズムと鍵サイズの移行

本推奨事項の第一部は、将来的にセキュリティメカニズムの強度を増加させるために多くのアプリケーションおよびプロトコルによって現在使用中のアルゴリズム及び鍵サイズからの移行についてのタイムフレームを提供している [\[SP 800-57 Part 1\]](#)。多くの場合、適切なセキュリティを提供するために要

求されるアルゴリズムと鍵サイズは、現在の実装では利用可能でないか、またはやり取りする必要がある利用者のコミュニティにおいて一様に利用可能ではない。新しいアルゴリズムまたは鍵サイズへの移行は、必ずしも即座に発生するわけではないが、システムの至る所で段階的なアップグレードを要求するだろう。例えば、システム所有者は、暗号モジュールをアップグレードする前に、システムの電子メールパッケージをアップグレードする必要があるかもしれない。したがって、ある一定の期間は、システムは、TDEA と AES128 暗号化を利用可能だが、TDEA を取り扱えない暗号モジュールのみの、電子メールパッケージを実行しているかもしれない。古い機能のサポートを継続しつつ、すべての要素がアップグレードされるまでの間は、システムの要素を新しい機能にアップグレードする必要があるだろう。

この移行期間の間、要素間の対話は、以下の方法の 1 つにおいて進めることができる：

1. 古いセキュリティメカニズム（例、使用されるセキュリティメカニズムをネゴシエートするようなプロトコルを用いて）の代わりに使用可能であるように、いくつかの手段は、所与のトランザクションにおいて新しいセキュリティメカニズムがすべての人々に利用可能となっていることを決定するために提供される。新しいセキュリティメカニズムが、トランザクションに関係するすべての人々に利用可能でないとき、古いセキュリティメカニズムが使用可能である。このアプローチは、ある人々が新しいメカニズムを持っているとき、彼らのトランザクションがより高いセキュリティレベルで保護されるという長所を持っている。欠点は、古いセキュリティメカニズムを用いたそれらのトランザクションが十分に保護されないことである；これは、同じ情報が異なるトランザクションにおいて 2 つまたはそれ以上の人々の間で異なる強度のセキュリティメカニズムを用いて送信されることがあるという可能性が挙げられる—事実上、新しいセキュリティメカニズム⁴によって提供されるより高いセキュリティ強度を無駄にしまう。
2. すべての要素がアップグレードされるまで、すべての要素は、古いセキュリティメカニズムを使用する；そのとき、システムは新しい機能へ直ちに移行する。このアプローチは、すべての要素が期限までにアップグレードされないために、期限を過ぎてすべての要素がアップグレードされるまでの期間すべての情報に対して不適切な保護を提供するという潜在的な問題を持っている。しかし、このアプローチは、同じデータが 2 つの異なるセキュリティレベル⁵で送信されないという長所を持っている。

第三部で議論されるアプリケーションとプロトコルのほとんどは、第一部に適合する利用可能なセキュリティメカニズムのアップグレードを要求する。以下のセクションは、適切なアップグレードが実施可能となるまで既存のメカニズムがどのように使用されるのが最善であるかについてガイダンスを提供する。組織とシステム管理者は、システム内のより強固なセキュリティメカニズムへの移行のためのアプローチを決定しなければならない。

⁴ より高いセキュリティレベルの利用者がその他の利用者が同じ情報を保護するためにより低いセキュリティレベルのメカニズムを使用しているかもしれないことを知らないときに問題となる。

⁵ 移行前に送信されたデータが移行後にまた送信されないことを想定する。

2 公開鍵基盤 (PKI)

2.1 説明

PKI は、公開鍵の配付のための最も共通の鍵管理アプローチである。SP 800-57 Part 1、*鍵管理における推奨事項、第一部：一般事項* [\[SP 800-57 Part 1\]](#) で記述されるとおり、公開鍵は、様々な保証を得た後のセキュリティサービスを確立するために使用される：完全性の保証、ドメインパラメータ有効性の保証（適切な場合）、公開鍵有効性の保証、およびプライベート鍵所有の保証。多くの場合、アプリケーションはこの鍵ペアに対応する利用者の同一性についても確立しなければならない。PKI において、基盤は使用者の同一性と、IPsec ([セクション 3](#))、トランスポート層セキュリティ ([セクション 4](#))、S/MIME セキュア電子メール ([セクション 5](#)) および Kerberos のいくつかのバージョン ([セクション 6](#)) を含めて PKI 使用可能なアプリケーションおよびプロトコルにおけるセキュリティサービスの強固な基礎を提供するために要求される保証を確立する。本セクションは、PKI ベースの鍵管理のためのガイダンスを提示する。より幅広くより詳細な PKI に関する情報については、[\[SP 800-32\]](#) を参照。

公開鍵証明書は、2 つの名前、利用者の名前と発行者の名前を発行者によって生成されたデジタル署名を用いて、公開鍵に結合する。利用者とは、証明書における公開鍵に対応するプライベート鍵を用いて許可された者である。発行者とは、利用者の同一性；公開鍵、関連するアルゴリズムと任意のパラメータの有効性；および対応するプライベート鍵の利用者の所有を検証した後、証明書を生成し署名するような信頼された第三者である。発行者は認証局 (CA) として知られている。多くの場合、CA は、サブジェクトの同一性の検証について登録局 (RA) に責任を負わせている。証明書は、依頼当事者として知られるその他の関係する者が PKI によって提供される保証および証明書生成処理を信頼したのちに、依頼当事者へ利用者の公開鍵を配付するために使用される。

CA は、一般に、ルート証明書 (時にはトラストアンカーとも呼ばれる) と呼ばれる、自己署名証明書を発行する；これは、CA によって発行された証明書を検証するためのアプリケーションとプロトコルによって使用される。CA 証明書は、多くのプロトコルとアプリケーションにおいてキーとなる役割を演じ、ルート証明書ストアと呼ばれるようなものにおいて一般的に維持される。Microsoft Windows オペレーティングシステムにおいて、さまざまな Microsoft プロトコルおよびアプリケーション、およびそれらを使用するための選択されるかもしれないその他のアプリケーションによって共有されるさまざまな目的でオペレーティングシステムによって維持されるルート証明書ストアがある。Apple オペレーティングシステムには同様の “Keychain” ファシリティがある。オペレーティングシステム間で可搬となるように意図された、いくつかのアプリケーションは、それらのルート証明書ストアを維持でき、その他のアプリケーション⁶とルート証明書ストアを共有できるような機能も持っている。

証明書は、一般に *証明書方針* に従って発行される。一般に方針は、発行する CA ウェブサイト上で見つけることが可能である。ある組織の方針が、例えば、少なくとも 2048 ビット RSA、2048 ビット DSA または 224 ビット楕円曲線暗号、および SHA-224 または SHA-256 のいずれかを用いる証明書のみを受け入れる場合、その公開鍵サイズが要求事項を満たすことを保証する唯一の実践的な方法は、通常、ルート証明書ストアがこれらのアルゴリズムと鍵サイズを要求するような証明書方針をもったルート証明書のみを含むことを保証することである。PKI を使用する現在のアプリケーションは、証明書で使用された公開鍵またはハッシュアルゴリズムの適切性をチェックしないこと以外は、アプリケーションのルートストアにおける証明書がルート証明書の下で発行されたこと、およびその後で失効されていないことを保証するためにチェックする。アプリケーションは数学的に正しい結果を計算するために特

⁶ さまざまな Mozilla ブラウザと電子メールクライアント、および Apache ウェブサーバが例である。Microsoft Internet Explorer、Outlook および Internet Information Server は、すべてウィンドウズルート証明書ストアを使用する；Apple Safari と Mail は、Keychain を使用し；Mozilla FireFox、Thunderbird および SeaMonkey はすべてそれら自身のルート証明書ストアを使用し、Mozilla Network Security Services (NSS) ユーティリティからのルート証明書ストアを共有可能である。

定の鍵を単に使用する。したがって、ルート証明書ストアを正しく設定することは、鍵管理における重要なステップである。

ルート証明書ストアが置かれる場所およびそれぞれのアプリケーションとプロトコルのためにどのように管理されるかについての詳細は、本推奨事項の適用範囲を超えている。しかし、通常は、ブラウザアプリケーションにおける証明書ストアを閲覧し、管理するためのメニューがあるが、それぞれの製品アップデートとともに変更の対象である。システム管理者によって集中管理のためのオペレーティングシステムまたはアプリケーションにおけるユーティリティと機能もあるかもしれない。ブラウザまたはその他のアプリケーションが承認されていない CA 証明書に遭遇するとき、エンドユーザは、永続的な信頼される証明書ストアにその証明書を追加するか、一時的に証明書を信頼するか、または証明書拒否しアプリケーションを閉じるか、を選択するよう促されるかもしれない。

最も一般的な証明書フォーマットは、X.509 バージョン 3 (X.509v3) 証明書 [RFC 5280](#) である。利用者と発行者の名称、および公開鍵に追加して、すべての X.509 証明者は、デジタル署名、有効期間 (開始日時と終了日時)、および公開鍵と署名と共に使用される暗号アルゴリズムを規定する識別子についても含む。X.509v3 証明書は、拡張機能を含んでいる ; CA は、通常、公開鍵がサポートするよう意図されている暗号操作、証明書発行を管理する方針、および失効された証明書をどこで見つけるか (即ち、証明書のステータスについての情報の権威ある情報源) を示すためのその証明書における標準の拡張を含んでいる。CA は、アプリケーションまたは利用者ドメインに特有の情報を含むような証明書における “private” 拡張を含むこともあるかもしれない。

依拠当事者は、証明書および有効な情報を提供するための証明書を発行した CA を信頼する個人または組織である ([附属書 A](#) を参照)。依拠当事者が証明書の公開鍵を使用する前に、彼はこの証明書を署名するために発行者によって使用された鍵が信頼できるかどうかを決定しなければならない。もっとも簡単な場合、依拠当事者は、発行者について知っており、その CA によって発行された証明書をすでに信頼することを決定している場合である。依拠当事者が直接信頼するような CA はトラストアンカーと呼ばれる。複数のトラストアンカーが承認されるとき、一連のトラストアンカーはトラストリストとして参照される。

ある場合には、依拠当事者は、信頼リストにある CA 以外の発行者によって署名された利用者証明書を処理することを望むであろう。この目標をサポートするため、CA は別の発行者名をその発行者の公開鍵に結合するようなクロス証明書を発行する。クロス証明書は、公開鍵がその他の証明書における署名を検証するために使用されるかもしれないという表明である。依拠当事者は、依拠当事者の信頼リストにないような CA によって発行されたにもかかわらず、利用者の公開鍵証明書が信頼できることを実証するような、証明パス (一連の証明書) を作ることができるかもしれない。すべての証明パスは、トラストアンカーから始まり、ゼロまたはそれ以上の中間証明書、および利用者の公開鍵を含む証明書で終わる。

全体のパスは、その証明書が失効していないこと、適切な方針の下で発行されたこと、およびそれぞれの公開鍵が指定された通りの用途に適していることを保証するために検査されなければならない。この処理は、パス検証として知られている。

上記のとおり、証明書自体は通常、証明書のステータスについての情報の権威ある情報源へのポインタを含んでいる。証明書のステータスについての情報は、2 つの標準メカニズムの 1 つを用いて提供されるかもしれない :

- ・ オンライン証明書ステータスプロトコル (OCSP) 応答 [RFC 6960](#)。OCSP レスポンドは、信頼されるシステムであり、依拠当事者からの要求への応答として、証明書毎ベースで、所与の CA の署名されたステータス情報を提供する。依拠当事者は、OCSP レスポンドのデジタル署名を検討することによって応答を認証できる。OCSP レスポンドが権威あるステータス情報を提供しているので、CA と OCSP レスポンドの間の公式な関係がなければならない (例. 契約)。

- 代替方法は、証明書失効リスト、CRL を用いることである。X.509 CRL は、失効した CA によって発行された証明書のリストを含み、それらがいつ失効したかを示し、失効した理由を含むかもしれない[RFC 5280]。期限切れでない証明書のシリアル番号が CRL 上にない場合、それはまだ有効である。CRL は、証明書のように、デジタル署名されるので、信頼されないシステムを通して配付することも可能である。最も一般的には、CRL は LDAP⁷ディレクトリまたはウェブサーバ経由で配付される。ウェブサーバによる CRL の配付は最近ではより一般的になっている。

多くの場合、PKI は、事業継続をサポートするために、鍵回復サービスも（回復サーバを用いて）提供している。鍵回復サービスは、暗号化データの平文が将来回復されるかもしれないことを保証するために鍵確立をサポートするようなプライベート鍵を保管する。これらのサービスは、喪失事象または暗号モジュールの故障において利用者に対して、または方針又は法的要求が存在するときに利用者の管理に対してプライベート鍵を提供することができる。サポートされるとき、このサービスは PKI 対応のアプリケーションから鍵管理の負担を取り除く。PKI は鍵回復サービスシステムを含むべきである。

本セクションは、異なる組織の利用者がさまざまなアプリケーションをサポートする必要があるときの汎用の PKI についてのガイダンスを提供する。大きな汎用の PKI について相互運用性は重要な検討事項である。あまり一般的ではないが、PKI は、より広範囲の相互運用性があまり重要でないような、小さな、閉鎖的な利用者コミュニティまたは単一のアプリケーションをサポートするために配備されるかもしれない。本セクションの要求事項は、大規模で汎用な PKI、連邦政府向け PKI のようなものに焦点を当てている。相互運用性をあまり要求しない PKI について、これらの要求事項は、それらのシステム内での適切性のために評価されるべきである。一般に暗号アルゴリズムと鍵サイズの標準はすべての PKI によって満たされるべきである。

2.2 セキュリティおよび適合性の問題

2.2.1 推奨される鍵サイズとアルゴリズム

以下の表 2-1 は、PKI 利用者および基盤の要素によって使用される鍵ペアの推奨される鍵サイズについて要約している。PKI は、否認防止サービスを提供するような、署名用プライベート鍵または署名検証用公開鍵（第一部で定義されるとおり）を参照するため、用語 *デジタル署名用鍵* を使用する。用語 *認証用鍵* は、第一部で定義されたとおり、認証用プライベート鍵または認証用公開鍵を参照するため、PKI によって使用される。デジタル署名用鍵と認証用鍵の両方は、デジタル署名アルゴリズムと共に使用されることに注意すること。

本表における日付は、第一部にあらわれるものと一貫している：

- デジタル署名用鍵は、上記定義のとおりである。
- 鍵確立用鍵は、鍵共有または鍵配送を提供するために使用される非対称鍵ペアである。そして
- CA および OCSP レスポンダ署名鍵は、証明書を署名し、検証するために使用される非対称鍵ペアである。

表 2-1 で規定される承認されたアルゴリズムと鍵サイズ、および証明書有効期限は 2 つの異なるものである。これらのアルゴリズムと鍵サイズは 2013 年を超えて使用することが承認されている。しかし、デジタル署名または鍵確立証明書は、いつでも組織のセキュリティ方針によって期限切れとされるかもしれない。

⁷ **Lightweight Directory Access Protocol (LDAP)** は、誰かが、インターネット上または企業内イントラネットであろうとなかろうと、ネットワーク上のファイルやデバイスのような、組織、個人、及びリソースを探すことを可能にする、ソフトウェアプロトコルである。[RFC 4511] *Lightweight Directory Access Protocol (LDAP) : The Protocol* および [RFC 4512] *Lightweight Directory Access Protocol (LDAP) : Directory Information Models* を参照。

表 2-1：推奨されるアルゴリズムと鍵サイズ

鍵タイプ	アルゴリズムと鍵サイズ
認証のために使用されるデジタル署名用鍵 (利用者またはデバイス用)	RSA (2048bits) ECDSA (Curve P-256)
否認防止のために使用されるデジタル署名用鍵 (利用者またはデバイス用)	RSA (2048bits) ECDSA (Curves P-256 または P-384)
CA および OCSP レスポンダ署名鍵	RSA (2048 または 3072bits) ECDSA (Curves P-256 または P-384)
鍵確立用鍵 (利用者またはデバイス用)	RSA (2048bits) Diffie-Hellman (2048bits) ECDH (Curves P-256 または P-384)

DSA 2048 のような、いくつかの承認されたアルゴリズムと鍵サイズが相互運用性を高めるため、省略されていることに注意すること。上記表 2-1 に含まれる RSA および ECDSA は、PKI において広く配備されている。したがって、それらは、相互運用性を高めるため、使用するよう推奨されている。しかし、[\[FIPS 186-4\]](#) で規定されるとおり、DSA (2048 および 3072) は、要求されるセキュリティ強度が満たされる限り許容される。ECDSA については、楕円曲線のうち、上記の表 2-1 に列挙された 2 つの楕円曲線のみがデジタル署名用として PKI における使用が推奨されている[\[FIPS 186-4\]](#)。同様に楕円曲線 Diffie-Hellman (ECDH) は、楕円曲線 MQV よりもむしろ、鍵確立をサポートするために推奨されている。

表 2-1 は証明書に含まれる公開鍵の強度に焦点を当てているが、証明書自身のデジタル署名の強度は等しく重要である。署名セキュリティ強度は、さらには署名を生成するために使用されたプライベート鍵の強度に追加して、おそらくはパディングスキーム⁸も含めた、ハッシュアルゴリズムのセキュリティ強度についても反映している。以下の表 2-2 は、推奨されるアルゴリズム、鍵サイズ、ハッシュ関数、および署名する証明書のパディングスキームおよび CA による CRL、および OCSP レスポンダによる OCSP ステータスメッセージについて要約している。

表 2-2：CA および OCSP レスポンダのためのデジタル署名推奨事項

公開鍵アルゴリズムと鍵サイズ	ハッシュアルゴリズム	パディングスキーム
RSA (2048 または 3072bits)	SHA-256	PKCS #1 v1.5、PSS
ECDSA (Curve P-256)	SHA-256	N/A
ECDSA (Curve P-384)	SHA-384	N/A

RSA または Diffie-Hellman 公開鍵を含む利用者証明書は、RSA 署名アルゴリズムを用いて署名されるべきである。楕円曲線公開鍵を含む利用者証明書は、ECDSA を用いて署名されるべきである。

⁸ RSA は、PKI で使用される 2 つのパディングスキームを持っている：PKCS #1 v1.5、および PSS。ECDSA を用いて生成されるデジタル署名のセキュリティ強度はパディングスキームによって影響されない。

アルゴリズムと鍵サイズのすべての組合せが連邦政府情報の保護に適しているとは限らない。相互運用性を強化するため、利用者は、認証、署名及び鍵確立証明書をすべての公開鍵⁹についての補完的なアルゴリズムとともに取得すべきである。ほとんどの利用者について、署名用及び鍵確立用鍵は、同じ暗号強度を提供すべきである。以下の表 2-3 は、利用者鍵の望ましい組合せを示す。

対称鍵暗号については厳密には要求されないが、ブロック暗号が実際にはすべての PKI 実装および PKI 利用可能なアプリケーションにおいて使用される。ブロック暗号を用いるすべての構成要素は、AES-128 アルゴリズムをサポートしなければならない。古い実装をサポートするため、RSA 鍵を処理する要素は、3 つの鍵を使用するトリプル DEA [\[SP 800-67\]](#) をサポートすべきである。P-384 楕円曲線鍵および SHA-384 アルゴリズムをサポートする要素は、AES-256 をサポートしなければならない。

表 2-3 : 推奨されるアルゴリズムと鍵サイズのための推奨される組み合わせ

認証用鍵タイプ	署名用鍵	鍵確立用鍵
RSA 2048	RSA 2048	RSA 2048
RSA 2048	RSA 2048	Diffie-Hellman 2048
ECDSA P-256	ECDSA P-256	ECDSA P-256
ECDSA P-256	ECDSA P-384	ECDSA P-384
ECDSA P-384	ECDSA P-384	ECDSA P-384

2.3 調達ガイダンス

以下は、PKI のサポートにおいてどの製品を調達するかについての決定を行う責任者のためのガイダンスを提供する。

2.3.1 CA/RA ソフトウェアおよびハードウェア

1. CA および RA ソフトウェアがこれらのプロトコルのうち少なくとも 1 つをサポートすることを保証する：証明書管理プロトコル (CMP) [\[RFC 4210\]](#)、セキュアトランスポート経由の登録 (EST) [\[RFC 7030\]](#) および暗号メッセージ文法を用いた証明書管理 (CMC) ; [\[RFC 5272\]](#) を参照。
2. すべての CA が [\[RFC 5280\]](#) に適合するような証明書および CRL の生成をサポートすることを保証する。(証明書および CRL 拡張に関する具体的な要求事項が以下に詳述される。)
3. CA が利用者に対して複数の証明書、および Extended Key Usage 拡張を含め、Key Usage 拡張を主張

⁹ 一般に、プロトコルとアプリケーションは 1 つの数学的ファミリーからの暗号アルゴリズムを使用するように設計される。例えば、ECDSA デジタル署名を持つ証明書に遭遇するアプリケーションは、鍵確立サービスのために楕円曲線 Diffie-Hellman を使用することが期待される。ECDSA 証明書 (即ち、デジタル署名を検証するために使用される ECDSA 公開鍵を含む証明書)、および RSA 鍵確立証明書 (即ち、鍵確立のために使用される RSA 公開鍵を含む証明書) を取得する利用者は、例えば、1 つのアプリケーションでその鍵を一緒に使用できないことが判るかもしれない。証明書のその他の組合せは、一般に使用される (表 2-3 を参照)。アプリケーションやプロトコルにおいて鍵と一緒に使用可能であることを保証するために補完的であるような認証、署名、および鍵確立証明書を利用者が取得するほうがよい。

するようすべての証明書を発行できることを保証する。

4. CA が CRL Distribution Point 拡張を含めることができることを保証する。
5. CA が CRL の場所を規定するために HTTP URL を含めることをサポートすることを保証する。
6. CA は、CRL の場所を規定するため、LDAP を含むことをサポートするべきである。
7. CA が Authority Information Access 拡張において権威ある OCSP レスポンダを規定することができることを保証する。
8. それぞれの PKI は、発行する証明書および CRL に現れるような証明書拡張を識別するような、自分自身の証明書プロファイル¹⁰を持つ。CA が適切なプロファイルにおけるすべての必須の拡張を生成できることを保証する。連邦政府機関により所有される、または連邦政府機関を代行して運用される CA について、以下の具体的なガイダンスが適用される：
 - a) 連邦政府特有の方針を実装するような CA が、政府機関のプロファイルおよび連邦政府 PKI 証明書プロファイル [\[FPKI PROF\]](#) を満たすような証明書および CRL を生成できることを保証する。
 - b) 一般の方針フレームワーク [\[COMMON\]](#) を実装するような CA が共有サービス証明書および CRL プロファイル [\[COMMON PROF\]](#) を満たす証明書および CRL を生成できることを保証する。
9. CA は、証明書および CRL における “private” 拡張¹¹を含むことをサポートするべきである。
10. デジタル署名証明書および CRL のための以下のアルゴリズムのうち少なくとも 1 つをサポートすることを保証する：PKCS #1 v1.5 パディングを伴う RSA ; PSS パディング [\[RFC 3447\]](#) を伴う RSA、DSA または ECDSA。フレキシビリティを最大化するために、CA は、RSA と ECDSA¹²をサポートするべきである。
11. ルート鍵が破損、破壊または喪失し、CA の喪失したステータスを単に回復するよりも、バックアップルート鍵を用いて CA を再構築する必要があるような事象における CA の再構築をサポートするため、CA がバックアップおよびアーカイブ機能を含むことを保証する。CA は、いつ証明書が発行され、失効されるかを権威の下に確立するためにバックアップおよびアーカイブ機能を含めるべきである。
12. CA/RA 要素は、調達場所から CA または RA の物理的な場所への連続した責任追跡性を提供するような、管理された方法で出荷され、または配付されることを保証する。

2.3.2 OCSP レスポンダ

1. OCSP レスポンダが RFC6960、*Online Certificate Status Protocol* [\[RFC 6960\]](#)に適合することを保証する。
2. OCSP レスポンダが、署名された要求と署名されていない要求の両方を処理することができ、また

¹⁰ このプロファイルはしばしば、明示的に文書化されるが、証明書方針を通して暗黙的に規定されてもよい。

¹¹ プライベート拡張は、組織によって彼ら自身のユニークな要求事項を満たすために定義される。重要でないプライベート拡張は証明書または CRL の相互運用性に影響を与えないことに注意すること。

¹² 運用中の CA において証明書および CRL を署名するために使用されるアルゴリズムは、使用される暗号モジュール及び OCSP レスポンダのソフトウェアの両方に依存する。選択されたアルゴリズムは、サポートされる一連のアルゴリズムの両方において現れなければならない。

要求を作成する依頼当事者の名称を含むか省くかのいずれかのような要求を処理できることを保証する。

3. OCSP レスポンダが、[RFC 5019](#) で規定されるとおり、証明書ステータス要求の処理および非エラーステータスの応答の生成が可能であることを保証する。
4. サポートされる場合、OCSP レスポンダは、証明書を署名するために使用されるアルゴリズムと鍵サイズを用いて OCSP 応答を署名するべきである。OCSP レスポンダがデジタル署名応答メッセージの以下のアルゴリズムのうち少なくとも1つをサポートすることを保証する：PKCS #1 v1.5 パディングを伴う RSA ; PSS パディングを伴う RSA、DSA または ECDSA。サポートされるアルゴリズムは、ステータスが問合せされた証明書に署名した時に対応する CA によって使用されたアルゴリズムを含むべきである。CA による将来のアルゴリズム移行をサポートするため、OCSP レスポンダは RSA と ECDSA¹³をサポートするべきである。

2.3.3 暗号モジュール

1. CA、鍵回復サーバ、および OCSP レスポンダのための暗号モジュールは FIPS 140-2 レベル 3 またはそれ以上 [FIPS 140-2](#) を満たすような認証済のハードウェアモジュールであることを保証する。
2. RA の暗号モジュールは、FIPS 140-2 レベル 2 またはそれ以上 [FIPS 140-2](#) を満たすような認証済のハードウェア暗号モジュールであることを保証する。
3. 依頼当事者および利用者の暗号モジュールは、FIPS 140-2 レベル 1 またはそれ以上 [FIPS 140-2](#) を満たすような認証済であることを保証する。

2.3.4 鍵回復サーバ

1. PKI が鍵確立（即ち、証明書が鍵配送または鍵共有を含む）をサポートする場合、PKI は、鍵回復メカニズムを含むべきである。
2. 実装は、自動化、利用者起動による鍵回復をサポートするべきである；組織による鍵回復についてもサポートされるべきである¹⁴。

2.3.5 依頼当事者のソフトウェア

1. 依頼当事者のパス検証
 - a) 依頼当事者の実装は、RFC5280 適合のパス検証 [RFC 5280](#) を使用することを保証する。
 - b) 1つの組織の外の相互運用性が要求される場合（例、1つの連邦政府機関）、パス検証モジュールは、X.509 パス検証の NIST 推奨事項 [\[X.509 Path\]](#) で規定されるとおり、エンタープライズパス検証モジュール（PVM）の要求事項に適合するべきである。
 - c) 組織を越えて相互運用性が要求される場合、X.509 パス検証の NIST 推奨事項 [\[X.509 Path\]](#) ドラフトで規定されるとおり、パス検証モジュールは、ブリッジ利用可能な PVM の要求事項に適合するべきである。
 - d) 依頼当事者の実装は、証明書のステータスについて CRL または OCSP をサポートすることを保証する、またその両方をサポートするべきである。

¹³ CA と同様に、運用中の OCSP レスポンダにおいて応答に署名するために使用されるアルゴリズムは、使用中の暗号モジュールと OCSP レスポンダのソフトウェアの両方に依存する。選択されたアルゴリズムはサポートされる一連の複数のアルゴリズムの両方に現れなければならない。

¹⁴ 組織の鍵回復は、パフォーマンスよりもセキュリティとプライバシーを強調するべきである。組織による利用者の鍵の回復のためのデュアルコントロールが強く推奨される。

2. 証明書パスの構築

- a) 依拠当事者の実装は、証明書パスを構築できることを保証する。
 - b) 最小限、実装は、**http** ベースの証明書検索をサポートするべきである。
 - c) 依拠当事者の実装は、利用者の証明書内に規定される場所と同様に、LDAP を用いて組織の指定されたローカルディレクトリから CA 証明書および CRL を取得することもできるべきである。
3. 1つの運用中の PKI 内（例. 会社または政府機関をサポートする PKI）で活動するような依拠当事者は、トラストアンカーCA の階層的に下位にある CA によって発行された利用者証明書のパスを知ることができるべきである。
 4. その他の組織からの証明書を受け入れる依拠当事者は、非階層的な PKI におけるパスを知ることができるべきである。

2.3.6 クライアントソフトウェア

1. クライアント実装は、異なる暗号サービスをサポートするためそれぞれのエンドユーザの複数のプライベート鍵と証明書をサポートすることを保証する。例えば、クライアント実装は、デジタル署名をサポートする証明書にある公開鍵と関連するプライベート鍵と鍵確立をサポートする証明書にある公開鍵と関連するプライベート鍵のサポートとそれらの間の区別をするべきである。
2. クライアントの暗号モジュールは、FIPS 140-2 レベル 1 またはそれ以上 [\[FIPS 140-2\]](#) で認証済であることを保証する。
3. クライアント実装は、組織の CA¹⁵によってサポートされる証明書管理プロトコルをサポートするべきである。

2.4 システムの設置者／管理者のための推奨事項

システム設置者及び管理者は、PKI の確立に責任を持ち、日々の運用に関連する作業に責任を持つ人（または人々）である。システム管理者は、エンドユーザが訓練されること及び組織のセキュリティ方針が実施されることを保証しなければならない。

2.4.1 証明書発行

1. CA は、証明書が承認された鍵サイズを持つ公開鍵、有効なドメインパラメータ（適切であれば）、及び承認されたアルゴリズムを規定することを保証するよう構成されなければならない。
2. 最大限の相互運用性のため、CA と利用者は、デジタル署名と鍵配送のために RSA 鍵ペアを使用するべきである。
3. 最大限のセキュリティとパフォーマンスのため、CA と利用者は、デジタル署名と鍵配送のために楕円曲線を使用するべきである。
4. 証明書または CRL に署名するとき、CA は、表 2-2 で規定された組み合わせの、署名アルゴリズム、

¹⁵ 鍵と証明書がスマートカードに保管される場合、かつすべてのアップデートが RA で実行される場合、利用者の実装は、証明書管理プロトコルをサポートする必要はない。

ハッシュ関数およびパディングスキームを用いて、デジタル署名を生成しなければならない。

5. デジタル署名証明書について、CA は、その証明書におけるサブジェクト公開鍵のセキュリティ強度と等しいかまたはそれ以上のセキュリティ強度を持つ、デジタル署名処理（即ち、署名アルゴリズム、ハッシュ関数と鍵）を用いて証明書に署名しなければならない。鍵確立証明書について、CA は、その証明書¹⁶におけるサブジェクト公開鍵のセキュリティ強度よりも少ないセキュリティ強度を持つ、デジタル署名処理を用いて証明書に署名してもよい。
6. 鍵ペア生成：
 - a) 利用者は、彼ら自身のデジタル署名用鍵ペアを生成するべきである。
 - b) 鍵確立用鍵ペアは、利用者によって、または利用者に代わり PKI によって生成されてもよい；要求される場合、利用者の鍵ペアを生成する PKI は、鍵リカバリを許可するために鍵確立プライベート鍵のコピーを保持するかもしれない。
 - c) CA は、証明書を発行する前にすべての鍵ペアについて保有の証明を実行するべきである。
7. CA は、証明書を発行する前に公開鍵有効性の保証を得なければならない。
8. Key Usage 拡張。
 - a) すべての発行された証明書は、Key Usage 拡張を含まなければならない。
 - b) Key Usage 拡張は、プライベート鍵の受け入れを 1 つの暗号機能に制限しなければならない：利用者／エンティティ認証およびコミットされたデータの検証、または鍵確立のいずれか。
9. 全ての証明書は、ステータス情報の検索をサポートするため CRL Distribution Point 拡張を含まなければならない。
10. OCSP レスポンドがサポートされる場合、証明書は、Authority Information Access 拡張における適切な URL を含まなければならない。
11. 証明書は、それが有効期限切れとなる前に更新されるべきであり、ドメイン名または組込まれた電子メールアドレスのような証明書の内容に変更がある場合には置換されるべきである。

2.4.2 証明書失効要求

1. CA は、現実的である場合、失効処理を自動化するよう構成されるべきである：
 - a) CA は、電子的に失効要求を認証し処理するよう構成されるべきである。
 - b) CA が関連する鍵ペアの利用者または RA により提出されたデジタル署名された要求を認証できる場合、その要求は手動による仲介なしに取り扱われるべきである。
2. RA は、利用者または組織に代わってデジタル署名された失効要求を提出するよう構成されるべきである。

¹⁶ 鍵確立に使用される公開鍵証明書は、2つの鍵を含む：データを保護するような対称鍵を確立するために使用されるサブジェクト（鍵確立）公開鍵、及び証明書に署名するために使用される認証局の署名鍵。CA の署名鍵は、鍵確立証明書が無効となるまでの間のみ、セキュアである必要があるが、サブジェクト（鍵確立）公開鍵は、鍵確立証明書の有効期限後も続くかもしれないような、データがセキュアでなければならない期間はセキュアである必要がある。証明書の寿命の間に CA 署名鍵はセキュアである限り、証明書はセキュアにアーカイブされ、証明書の有効期限後の CA 署名鍵が破られることはサブジェクト（鍵確立）公開鍵の有効性または（サブジェクト公開鍵）が提供できるセキュリティに影響しない。例えば、サブジェクト（鍵確立）公開鍵のセキュリティ強度が CA の署名鍵のセキュリティ強度よりも大きい場合、サブジェクト公開鍵が署名された後に署名鍵が破られることはその公開鍵のセキュリティに影響を与えない。したがって、鍵配送または鍵確立サブジェクト公開鍵が、鍵共有または鍵配送公開鍵を含む証明書を署名するために使用される CA 鍵よりも強いことは、受け入れ可能である。

2.4.3 証明書失効リストの生成

1. 最大限の相互運用性のため、すべての CA は、完全な CRL を生成するよう構成されるべきである。完全な CRL は特定の CA によって発行された有効期限切れでない証明書で失効されたすべてのリストであるような 1 つの CRL である。
2. 大規模なコミュニティにサービスを提供する CA は、完全な CRL に追加して CRL 配付ポイントを生成するべきである。それぞれの CRL 配付ポイントは所与の CA についての失効した証明書のサブセットを列挙している。CRL 配付ポイントによって網羅される証明書の数は、配付ポイント CRL が管理不能な大きさに増大しないことを保証するため、最大でも 25 万件までに限定するべきである。

2.4.4 証明書と CRL の配付のための PKI リポジトリ

1. PKI は、要求者の認証なしに要求者に対して証明書と CRL を提供するよう構成されるべきである。
2. PKI リポジトリは、リポジトリによって配付される一連の証明書と CRL を改変するためには認証されたアクセスを要求されるように構成されなければならない。
3. 最小限、リポジトリは HTTP バージョン 1.1 または LDAP バージョン 3 のいずれかのインタフェースをサポートしなければならない。
4. 最大限の相互運用性のため、HTTP と LDAP の両方がサポートされるべきである。
5. 可用性の最大化するためのリポジトリの複製（例、ディレクトリシャドウイングまたはウェブサーバ複製を通して）は、考慮されるべきである。
6. PKI リポジトリは、対応する PKI によってまたは PKI へ発行されたすべての CA 証明書を含むべきである。
7. PKI リポジトリは現在のすべての CRL を含まなければならない。

2.4.5 OCSP レスポンダ

連邦政府機関について、OCSP レスポンダの詳細な構成ガイダンスが米国連邦政府 PKI における OCSP レスポンダのドラフトガイダンス¹⁷で規定される。

1. 最大限の相互運用性が要求される場合は：
 - a) OCSP レスポンダは、要求が署名されることを要求してはならない、かつ証明書のステータス情報が提供されるような依拠当事者に限定してはならない。
 - b) レスポンダは、OCSP basic 応答を生成しなければならない、また応答は critical 拡張を含んではならない。
2. 相互運用性要求事項が閉鎖的なコミュニティに限定される場合：
 - a) OCSP レスポンダは、署名された要求を要求するかもしれない、またはそのコミュニティの外のエンティティからの要求を拒否するかもしれない。
 - b) OCSP 応答メッセージは、対称コミュニティ内で既知の private 拡張を含むかもしれない。

¹⁷ <http://cio.nist.gov/esd/emaildir/lists/pkits/doc00000.doc> にて利用可能である。

2.4.6 バックアップとアーカイブ

1. ステータス情報の可用性を維持するため、CA は十分な情報が災害の後で、CA を再構築するためのセキュアな場所に保管されることを保証しなければならない。
2. CA は、証明書がいつ、どの権威の下で発行されたかを確立するため十分な情報をアーカイブすべきである。
3. 一般的なルールとして、監査ログは、要求される限り対応する証明書に沿って、維持されるべきである。
4. 利用者署名検証公開鍵は、要求される限り対応する証明書に沿って、アーカイブされるべきである。

2.4.7 依拠当事者の統合と設定

1. パス発見の要素は、パス発見を有効化し、ステータス情報の検索を要求するよう構成されなければならない。
2. ステータス情報は、CRL と OCSP の両方のフォーマットで受け入れられるべきである。
3. 依拠当事者の実装は、最小限の受け入れ可能なトラストアンカーを可能であれば承認するように構成されなければならない。
4. 民間と政府期間、および政府機関同士のアプリケーションについて、連邦政府機関は、Common Policy Root CA またはトラストアンカーとして Common Policy Root CA または連邦政府ブリッジとクロス認証された政府機関 CA のいずれかを使用するべきである。
5. 限定セキュリティ要求事項（例、[OMB 04-04](#)）で規定のとおりレベル 2 e-Authentication 要求事項）および高い相互運用性要求事項を伴う市民と政府機関のアプリケーションについて、政府機関は、COTS（訳注：市販の）製品において提供された、事前インストールされたトラストアンカーを使用するかもしれない。
6. パス検証モジュール：
 - a) エンドユーザアプリケーションおよび最低限のセキュリティ要求事項について、パス検証モジュールは任意の検証パスを受け入れるように構成されるべきである。
 - b) より重要なセキュリティ要求事項を伴うシステム（例、レベル 3 またはレベル 4 の e-Authentication を満たすような PKI を用いるシステム）について、パス検証モジュールは適切な方針の下で有効であるようなパスのみを受け入れるよう構成されるべきである。

2.5 利用者ガイダンス（加入者）

PKI において、サブジェクトは公開鍵と関連付けられた利用者の同一性である。サブジェクトは、人またはデバイスであるかもしれない。本セクションの目的として、用語“利用者”は公開鍵と関連付けられた人、または公開鍵に関連付けられたデバイスの管理者のいずれかを意味する。

1. 利用者は、デジタル署名及び認証のため、自分自身の鍵ペアを生成するべきである。
2. 利用者は、鍵確立のため、自分自身の鍵ペアを生成するかもしれない、または鍵確立用鍵ペアは信頼された情報源からインポートされるかもしれない。
3. 利用者は認証子（例、PIN またはパスワード）を保護しなければならない。
4. 利用者は、認証子または暗号モジュールが盗難に合い、複製され、または危殆化したと彼らが

信じる場合、それらの証明書の失効を要求しなければならない。

5. 利用者は、連邦政府機関の方針と手順に従って管理されることがない限り、証明書が有効期限切れとなった後で“古い”鍵ペアの処分を管理しなければならない。
 - a) 署名プライベート鍵は、対応する証明書が有効期限切れとなった後、破壊されるべきである。
 - b) 鍵確立プライベート鍵は、対応する証明書が有効期限切れとなった後、破壊される必要はない。利用者は、この鍵を用いて確立したすべての対称鍵が回復されるまたはさもなければ保護（例. 異なる鍵の下で暗号化することによって）されるまでは、鍵確立プライベート鍵を破壊するべきでない。鍵確立プライベート鍵の早計な破壊は、加入者の平文データの回復を妨げるかもしれない。
6. 利用者は、CRL における証明書を拒否する前にすべての CRL を検証しなければならない。

3 インターネットプロトコルセキュリティ (IPsec)

3.1 説明

IPsec は、ネットワーク層でのインターネット通信をセキュアにするためのプロトコルスイートであり、インターネットプロトコル (IP) 内で動作する。両者に対して鍵材料を共有することを要求して、遠距離通信社または旅行者が彼らのビジネスネットワークへのセキュアなアクセスを得ることを可能にするような、仮想化プライベートネットワーク (VPN)¹⁸ を構築するためにしばしば利用される。IPsec は、インターネットプロトコルのバージョン 4 と 6 の両方のための暗号的セキュリティ機能を提供する。

IPsec は、それぞれのメッセージにおける IP ヘッダの後ろに 2 つの特別な IPsec ヘッダのうちの 1 つを挿入することで動作する。認証ヘッダ (AH) は、完全性保護を提供する。カプセル化セキュリティプロトコル (ESP) ヘッダは、機密性および/または完全性保護を提供する。以下、用語 AH と ESP は、AH および ESP ヘッダを用いるメッセージの略記として、それぞれ使用される。ESP と AH の両方ともデータ作成者の認証、およびオプションでリプレイ保護を提供する。AH は IP ヘッダおよび IP ヘッダに続くデータを保護する。ESP は、あるパケットに直接適用されるとき (即ち、トランスポートモードにおいて)、データを保護するが、IP ヘッダを保護しない。しかし、トンネルモードでの ESP (挿入された新しい IP ヘッダを伴う) は、元の IP ヘッダを保護する。さらに自動化鍵を持つ ESP は、トランスポートまたはトンネルモードのいずれかにおいて、IP ヘッダの発信元及び宛先アドレスを保護する。AH 処理は不必要な複雑性をもたらし、かつ ESP は等価な機能を提供できるので、AH の使用は推奨されない。

IPsec には 3 つのバージョン¹⁹がある。すべての新しいシステムは IPsec-v3²⁰を実装するべきである、以前のバージョンでは見つからない数多くの強化がなされている。しかし、IPsec-v2 は、廃止された²¹という事実にも関わらず、まだ膨大な現行システムにおいて実装されている。

鍵管理方法の 2 つのクラスが IPsec のために規定されている: 手動鍵と自動化鍵。手動鍵は IPsec 保護が適用される通信における当事者による (規定されていない方法での) 合意および使用される対称鍵が含まれる。これは、セキュリティソリューションのスケラビリティを大幅に制限し、かつ規定されないやり方で行われる鍵再作成を要求するという大きな欠点を持つ。セキュリティアソシエーション (SA、即ちセキュアに通信するためにそれぞれのエンティティがセキュリティサービスを利用しようとする方法について記述するような 2 つ以上のエンティティ間の関係) およびその秘密鍵は、SA が有効期限切れとなるような場合、最大許容容量のトラフィックを使い切った、またはその鍵が危殆化したような場合、容易に新しいものに置き換えることができない。

自動化鍵の取扱いを利用するため、IPsec 保護トラフィックを交換する前にピア間での自動化されたネゴシエーションは、適用される IPsec 保護と使用される共通鍵を決定する。同じ方法がその SA (例. 鍵再作成すること) を維持、削除、または再ネゴシエートするために使用されることが可能である。このアプローチは、その他のセキュリティメカニズムからの鍵管理メカニズムの分離を許す、したがってその他のセキュリティメカニズムを改変しなければならないことなしに代替りの鍵管理方法の使用を促進することを許容する。

望ましい自動化鍵の取扱い方法は IKE、これは IPsec と共に使用するために特別に設計された

¹⁸ SP 800-77 を参照、*IPsec VPN へのガイド* [\[SP 800-77\]](#)。

¹⁹ IPsec-v3 および IPsec-v2 の名前は一般的に受け入れられたものではない; 3 つの用語は本文書において、要求事項をより理解しやすいものとするために使用されている

²⁰ IPsec-v3 は、[\[RFC 4301\]](#)、[\[RFC 4302\]](#)、[\[RFC 4303\]](#) および [\[RFC 4835\]](#) において規定されている。

²¹ IPsec-v2 は、[\[RFC 2401\]](#)、[\[RFC 2402\]](#) および [\[RFC 2406\]](#) において規定されている。

Internet Key Exchange プロトコルである。IKE は、2つの IKE ピア間の自動化されたセキュアチャネル経由で IPsec の必須の鍵材料を生成する。2つのバージョンの IKE が使用されている：IKEv1 ([\[RFC 2407\]](#)、[\[RFC 2408\]](#) および [\[RFC 2409\]](#)) と IKEv2 [\[RFC 5996\]](#)；両方のバージョンは、相互認証を実行し、セキュリティアソシエーションを確立し維持する。SA は、規定された時間またはトラフィック容量について有効となる。IKEv1 は、これが廃止されたという事実にも拘らず、多くの現行システムにおいてまだ実装されている。これらの 2つのバージョンの IKE は、相互運用可能ではない。IKEv2 は、IKEv1 よりもより信頼可能であり、より効率がよいように設計されたものである；したがって、IKEv2 が使用されるべきである。

表 3-1 は、IPsec バージョン 2 と 3 の IETF 資料への参照を提供する。

表 3-1：IPsec の参照サマリ

バージョン	セキュリティアーキテクチャ	プライバシー	認証	自動化鍵管理
IPsec-v2	RFC 2401	RFC 2406	RFC 2402、 RFC 2406	RFC 2407、RFC 2408、 RFC 2409
IPsec-v3	RFC 4301	RFC 4303	RFC 4302、 RFC 4303	RFC 5996

IPsecのセキュリティメカニズムは、任意の具体的な暗号アルゴリズムと結び付けられていない；事実多くのアルゴリズムと利用モードがIPsecでの使用についてIETF Requests For Comment (RFCs) に記述されている。しかし、これは、通常システム管理者が相互運用性を達成することを困難にするほど多くの選択肢があるという状況を作り出している。IPsec-v3における相互運用性を向上するため、2つの暗号スイート：VPN-A と VPN-Bが規定された[\[RFC 4308\]](#)。しかし、これらの2つの暗号スイートはNIST承認された暗号スイートではない。4つの追加の暗号スイートが[\[RFC 6379\]](#)で定義された：Suite-B-GCM-128、Suite-B-GCM-256、Suite-B-GMAC-128 および Suite-B-GMAC-256で、これらはNIST承認されたものである。

実装者は、[\[RFC 6379\]](#)で規定されたセキュリティアルゴリズム（即ち、事前に規定されたスイートのアルゴリズムの1つを選択するというよりもむしろ）の個別の選択が許容されるかもしれないが、利用者は、非標準のアルゴリズムグループ化を取ることが限られた相互運用性を引き起こすかもしれないことを周知されなければならない。しかし、IPsec が VPN という環境で使用される場合、セキュリティポリシーは集中監視されること、すなわち事前定義された暗号スイートの使用なしに相互運用性を保証することが可能である。現在 IETF アルゴリズムガイドランスが [\[RFC 7321\]](#) にある。

3.2 セキュリティおよび適合性の問題

3.2.1 暗号アルゴリズム

以下の表 3-2 は、IPsec 内で使用される暗号アルゴリズムの推奨事項を与える。IKE のために規定されるアルゴリズムは、IKE 自身のトラフィックを保護するために使用される。ESP と AH で使用されるアルゴリズムはデータトラフィックへの IPsec 保護を提供するために使用される；ESP または AH 内で使用されるこれらのアルゴリズムは、それらの用途をネゴシエートできなければならない。

以下の表 3-2 において、カラム 4 は、3つの IETF 要求事項レベルを用いることによって RFC で規定されるとおり、IETF 適合要求事項を列挙する：MUST、SHOULD,および MAY は、そのアルゴリズムが実

装される必要があるかどうかを示す。これらの要求事項レベルの定義と IETF 適合性言語に関するさらなる情報については、[RFC 2119](#)を参照。カラム 5 は、しかし、2つのレベルを用いる連邦政府適合性要求事項について述べている：必須とオプション。必須は、その機能が実装において使用可能であることが要求されることを意味し、オプションは、手法の実装が許可されていることを意味する。

表 3-2 : 暗号アルゴリズム推奨事項

プロトコル	暗号サービス	アルゴリズム /利用モード	IETF 要求事項 RFC 7321	連邦政府 要求事項
ESP	暗号化	TDEA (CBC モード)	MAY	オプション； 使用される場合、 TDEA は3つの異なる 鍵を使用されなければ ならない
ESP	暗号化	AES-128 (CBC モード)	MUST	必須
ESP	暗号化	AES-128 (カウンタモード)	MAY	オプション； 使用される場合、完 全性保護と共に使用 されなければならない い
ESP または AH	完全性保護	HMAC SHA1-96 (鍵強度は 112bits 以上 でなければならない)	MUST	必須
ESP または AH	完全性保護	HMAC SHA-256-128 (鍵強度は 112bits 以上 でなければならない)	RFC 4868 SHOULD	オプション
ESP	暗号化と 完全性保護	AES-128 (GCM モード)	RFC 4106 , RFC 4835	オプション
ESP	暗号化と 完全性保護	AES-128 (CCM モード)	RFC 4109 , RFC 4835 MAY	オプション
ESP または AH	完全性保護	AES-128 (GMAC モード)	RFC 4543	オプション
IKEv1 または IKEv2	暗号化	TDEA CBC モード	MAY	オプション； 使用される場合、 TDEA は3つの異なる 鍵を使用されなければ ならない
IKEv1 または IKEv2	暗号化	AES-128 CBC モード	MUST	必須
IKEv1 または IKEv2	疑似ランダム 関数	HMAC-SHA1	RFC 4109 , RFC 4307 MUST	必須
IKEv1 または IKEv2	疑似ランダム 関数	HMAC-SHA-256	RFC 4868 SHOULD	オプション
IKEv1 または IKEv2	Diffie-Hellman グループ 24	2048-bit MODP	RFC 5114	必須

プロトコル	暗号サービス	アルゴリズム /利用モード	IETF 要求事項 RFC 7321	連邦政府 要求事項
IKEv1 または IKEv2	Diffie-Hellman グループ 14	2048-bit MODP	[RFC 4109] , [RFC 4307] SHOULD	オプション
IKEv1 または IKEv2	楕円曲線 Diffie-Hellman	P-256 または P-384	[FIPS 186-4] , [RFC 5903] MAY	推奨 (Should)
IKEv1 または IKEv2	完全性	HMAC-SHA-1-96 (鍵強度は 112bits 以上 でなければならない)	[RFC 4109] , [RFC 4307] MUST	必須
IKEv1 または IKEv2	完全性	HMAC-SHA256-128 (鍵強度は 112bits 以上 でなければならない)	[RFC 4868] SHOULD	オプション
IKEv1	ピア認証	2048-bit RSA with SHA256	[RFC 4109] SHOULD	必須
IKEv1	ピア認証	DSA with SHA256	[RFC 4109] MAY	必須
IKEv2	ピア認証	2048-bit RSA with SHA256	[RFC 5996] MUST	必須
IKEv2	ピア認証	DSA with SHA256	[RFC 5996] MAY	オプション
IKEv1 または IKEv2	ピア認証	ECDSA-256 または ECDSA-384	[RFC 4754] MAY	推奨 (Should)

[\[RFC 4835\]](#) および [\[RFC 2410\]](#) において、ESP は、完全性保護が一切適用されない、または暗号化が一切使用されないことを意味する、NULL 完全性保護または NULL 暗号化のオプションをそれぞれ提供する；しかし、[\[RFC 4835\]](#) は、少なくともそれらのうちの 1 つが適用されることが必須であることを規定する。RFC では、NULL 完全性保護（しばしば、NULL 認証として参照される）は完全性の必要性なしに機密性が要求されるような状況における用途を意図している。NULL 完全性保護は、実際にはできないが、NULL 暗号化と共に使用されてはならない。ESP は、例えば、暗号化されていないパケットを送信する可能性がある（暗号化を NULL に設定する）が、例えば、HMAC-SHA1 を用いることによって、それらの完全性保護を要求するだろう。その反面、ESP は、AES-128 の CBC モードでパケットを送信することが可能であるが、完全性チェックを省略する（完全性保護を NULL に設定する）ことが可能である。

しかし、本推奨事項に適合するため、IPsec ESP 保護されたトラフィックは、HMAC-SHA1-96 のような完全性保護アルゴリズムの使用または AES-128 の GCM モードのような結合されたモードのアルゴリズムの使用のいずれかを通して、常に完全性保護されなければならない。したがって、暗号化された ESP は、NULL 完全性保護とともに使用されてはならない。

IPsec 保護されたトラフィックが完全性保護される時、完全性チェック値 (ICV) は、ESP ペイロードの完全性チェック値フィールド [\[RFC 4303\]](#)、または認証ヘッダ (AH) の完全性チェック値フィールド [\[RFC 4302\]](#) に保管される；このフィールドは IPsecv2 ([\[RFC 2406\]](#)、[\[RFC 2402\]](#)) における“認証データ”として参照される。

HMAC が完全性保護に使用される時、その ICV はたかだかハッシュ関数の出力値の長さとなる。例えば、HMAC-SHA1-96 と HMAC-SHA256-128 の ICV 値は、それぞれ 96 と 128 bit となる [\[RFC 4868\]](#)。

IETF は、まだ AES-XCBC-MAC ([\[RFC 3566\]](#)、[\[RFC 4434\]](#)、[\[RFC 7321\]](#)) のサポートを推奨しているが、

連邦政府による用途としては承認されていない。このような AES-XCBC-MAC は、完全性保護のために使用されてはならない。

アルゴリズムの新しいクラス、結合されたモードアルゴリズムと呼ばれるものが上記の表にある。それらは、IKE によってネゴシエート可能であり、IPsec-v3 暗号スイート内で使用可能である。これらのアルゴリズムは暗号化と完全性保護の両方を提供する。2 つの結合モードアルゴリズムが連邦政府用途として承認されました： AES の Galois/Counter Mode (GCM) [\[RFC 4106\]](#) と AES の counter mode with CBC-MAC (AES-CCM) [\[RFC 4309\]](#)。

AES-GMAC [\[RFC 4543\]](#) のような AES-GCM の派生モードもあり、暗号化は提供しないが完全性を提供する。このモードは ESP または AH ヘッダのいずれかの中で使用されるかもしれない。

AES-CCM、AES-GCM および AES-GMAC の ICV の最大長は、16 バイトである。実装は、これらの3つのアルゴリズムについて 16 バイトの ICV 長をサポートしなければならない ([\[RFC 4309\]](#)、[\[RFC 4106\]](#)、[\[RFC 4543\]](#))。

AES-GCM、AES-CCM、AES-CTR、および AES-GMAC は、手動で配付された鍵と共に使用されてはならない。AES-CTR または AES-CCM におけるカウンター値、または AES-GCM または AES-GMAC における ICV 値が同じ鍵を用いた複数のパケットに使用される場合、アルゴリズムの機密性メカニズムのセキュリティは、危殆化する。なぜなら、手動の鍵取扱いは、この制限に対する大きな問題を呈するため、手動で配付された鍵はこれらのアルゴリズムで使用されてはならない。IKE を用いる自動化された鍵の取扱いでは、それぞれのセキュリティアソシエーション内の 2 つのピアの秘密鍵を確立し、複製鍵の確立は非常に小さいものとなる。

以前の IETF ガイダンスにおいて、シングル DES の CBC モード [\[RFC 2405\]](#) の実装が必須であった；しかし、このアルゴリズムは情報を保護するために使用されてはならない。

IPsec は、セキュリティアルゴリズムの個別の選択を許容する。たとえば、表 3-2 と以下のガイダンス第一部 [\[SP 800-57 Part 1\]](#) を用いる実装者は、全体のセキュリティ強度が 112 ビットとなる IPsec スイートを構成するような以下のアルゴリズムを選択することが可能である：

- ESP 暗号化：AES の CBC モード
- ESP 完全性保護：HMAC-SHA1
- IKEv2 暗号化：AES の CBC モード
- IKEv2 疑似ランダム関数：HMAC-SHA1
- IKEv2 Diffie-Hellman グループ：2048-bit MODP
- IKEv2 完全性：HMAC-SHA1
- IKEv2 ピア認証：2048-bit RSA with SHA-256.

Suite-B-GCM-128 および Suite-B-GCM-256 のスイートは、両方とも [\[RFC 6379\]](#) で定義されている。現時点では、これらの暗号スイートは、幅広く使用可能または普及してはいない。

[\[SP 500-267\]](#) は、これらの暗号スイートのサポートはオプションと述べている。しかし、現実的であればいつでも、実装がこれらの暗号スイートをサポートするようなものを調達すべきである、またそれらは非常に高いパフォーマンスとセキュリティ強度が要求される場合はどこでも使用するよう選択されるべきである。上記の議論のとおり、AES-GCM は、暗号化と完全性保護の両方を提供するような結合されたモードのアルゴリズムである；ゆえにこれらのスイートは、完全性メカニズムが両方のスイートに NULL と [\[RFC 6379\]](#) に列挙されているという事実にも拘らず、両方のスイートで完全性を提供する。

3.2.2 追加の推奨事項

1. 認証ヘッダ (AH) は、IPsec バージョン 3 において使用されてはならない。
2. IKE は、鍵再作成機能とスケーラビリティを保証するため、自動化された鍵管理のために使用されるべきである。
3. 一度、ESP セキュリティアソシエーションが有効期限切れまたはもはや使用されなくなったなら、その ESP 暗号化鍵は、システムによって保護され続けなければならない、かつ保護するために使用されたデータが秘密を維持される必要がある間は、秘密を維持され続けなければならない。

3.3 調達ガイダンス

これらの推奨事項は、IP 総のセキュリティのための IPsec を含むようなセキュリティ製品を選択することに責任のある個人を支援するために書かれている。

1. 連邦政府内での使用のためのあらゆる IPsec システムは、自動化された鍵管理のために IKE の実装を含むべきである。
2. IPsec 実装がそれぞれの IPsec セキュリティ要素のための承認されたアルゴリズムを含んでいることを保証する。
3. IPsec 実装は、スイート B 暗号で使用されるアルゴリズムを含むべきである。

3.4 システムインストーラのための推奨事項

システムインストーラは、セキュリティのために IPsec を含む製品をインストールするような個人である。

1. IKE は、任意の IPsec システム内に自動化された鍵管理のために使用されるべきである。
2. NULL 暗号化は、完全性保護が要求されるが、機密性が要求されないときにのみ、採用されなければならない。
3. インストーラは、[セクション 3.2](#) で期待されたとおり、それぞれのセキュリティ要素について承認されたアルゴリズムを選択しなければならない。

3.5 システム管理者のための推奨事項

システム管理者は、IPsec を含むセキュリティ製品の日々の運用に責任のある個人である。

1. エンドユーザが適切に訓練され、組織のセキュリティ方針が実施されることを保証する。
2. 製品によって使用される鍵がその寿命期間中、保護されることを保証する。

3.6 エンドユーザのための推奨事項

エンドユーザは、セキュリティのために IPsec に信頼を置く製品を使用する個人である。

1. 製品を使用するための組織のセキュリティポリシーを周知され、従うよう訓練される。
2. 組織及びシステム管理者によって指示されるとおりにそれらのシステムを操作する。

4 トランスポート層セキュリティ (TLS)

本セクションは、SP800-52 改訂第1版、トランスポート層セキュリティ (TLS) 実装の選択、構成、および利用のためのガイドライン [\[SP 800-52\]](#) へ移動された。

5 セキュア／多目的インターネットメール拡張 (S/MIME)

5.1 説明

セキュア／多目的インターネットメール拡張 (S/MIME) は、セキュアなインターネットメールを送信し、受信するための一貫した方法を提供する。S/MIME は、一連の IETF RFC、即ち、[\[RFC 5751\]](#)、[\[RFC 5652\]](#)、[\[RFC 2045\]](#)、[\[RFC 2046\]](#)、[\[RFC 2047\]](#)、[\[RFC 4288\]](#)、[\[RFC 4289\]](#) および [\[RFC 2049\]](#) によって定義された一連の仕様である。S/MIME は、以下のような電子的メッセージアプリケーションのための暗号学的セキュリティサービスを提供する：

- ・ デジタル署名を用いる送信者の認証、
- ・ デジタル署名を用いるメッセージ完全性と作成者の否認防止、および
- ・ 暗号化を用いる機密性。

ゆえに、S/MIME は、デジタルな同一性の確立と共有の何らかの手段と同様に、デジタル署名の生成、ハッシュ値の生成、鍵の確立および電子メールの内容の暗号化のためのアルゴリズムのスイートを要求する。連邦政府の実装は、S/MIME 利用者の同一性を確立し、それらの同一性を、公開鍵証明書を通して利用者の公開鍵へ結合し、デジタル署名を提供し、内容の暗号化のために使用される鍵を提供し、またはメッセージ毎ベースで使用される対称鍵を確立するために、公開鍵基盤、特に X.509 PKI に信頼を置く。PKI に関する詳細な情報については、本推奨事項の[セクション 2](#)を参照。

保管された電子メールは、暗号化されたファイル完全性とネットワーク越しの送信に関連する鍵管理問題を包含する。ゆえに、1) 鍵ペアおよび／またはマルチキャスト（一人以上の受信者へ送信される）送信者と受信者の間の鍵管理関係を確立する、および 2) その電子メールを受信が復号できる、または完全性を検証できる必要がもはやなくなるまで、暗号化された電子メールに関連する鍵をセキュアに保管する、必要がある。

S/MIME は、電子メールに限定されない；ハイパーテキスト転送プロトコル (HTTP) のような、MIME プロトコルを採用するような転送メカニズムと共に使用されることが可能である。

5.2 セキュリティおよび適合性の問題

S/MIME 製品は、セキュリティ機能の異なる組合せと様々な暗号アルゴリズムで実装されることが可能である。送信者と受信者は、異なる機能を持っているかもしれない、異なる強度のアルゴリズムで保護されたメッセージを送信しているかもしれない。これは、数多くの相互運用性問題を引き起こす可能性がある。セキュアな電子メールを使用する連邦政府機関のクライアントは、以下について実行できなければならない：

- ・ 署名メッセージを送信及び受信する、
- ・ 暗号化メッセージを送信及び受信する、
- ・ 署名及び暗号化されたメッセージを送信及び受信する、
- ・ 署名された受領書を要求、送信および処理する、そして
- ・ セキュアな電子メールリストクライアントからのメッセージを処理する（必要に応じて、受領書の抑制、および必要に応じて、リスト受信者の非開示）。

さらに、連邦政府システムは以下について実施しなければならない：

- ・ FIPS 140-1 または FIPS 140-2 認証済 [\[FIPS140-2\]](#) であるような暗号モジュールを活用する、

- ・ 暗号学的な暗号スイート 1（以下の表 5-1 を参照）をサポートする、そして
- ・ 連邦政府 PKI X.509 証明書および CRL 拡張プロファイルに適合するような X.509 証明書をサポートする。

連邦政府クライアントは、[\[RFC 5035\]](#)²²で記述されたとおり、セキュリティラベル付きの電子メールを送信および処理することができ、また署名証明書属性を通して送信者の証明書を彼らの署名にセキュアに結合することができるべきである。

最も広く受け入れられている、標準の S/MIME プロファイルは、[\[RFC 5751\]](#) である。このプロファイルの機能のサポートで使用可能なすべての暗号アルゴリズムが連邦政府情報の保護に適切であるわけではない。S/MIME 仕様は、個別のアルゴリズムの選択を許容する。しかし、多くの暗号スイートは、アルゴリズムの特定の組合せを定義するよう規定されている。連邦政府組織は、鍵確立とメッセージ送信に S/MIME 実装における承認されたアルゴリズムを使用しなければならない。表 5-1 から 5-3 は、連邦政府の情報および情報システムを保護するために使用してもよいさまざまな暗号スイートを規定する（[\[SP 800-49\]](#)および[\[RFC 6318\]](#)に基づき）。以下の表に掲載された任意のアルゴリズムが使用可能であり、表示されたもの以外の組合せで連邦政府情報を保護することは、第一部および[\[SP 800-131A\]](#) で与えられるセキュリティ強度の期間の制限に従って、使用されてもよい。

表 5-1 : 暗号スイート 1

メカニズム	ガイダンス
デジタル署名	DSA 鍵サイズ 2048bits 以上 [FIPS 186-4]
ハッシュ	SHA-256 [FIPS 180-4]
鍵共有	Diffie-Hellman 鍵サイズ 2048bits 以上 [SP 800-56A]
暗号化	AES-128 CBC モード [FIPS 197] および [SP 800-38A]

表 5-2 : 暗号スイート B、レベル 1*

メカニズム	ガイダンス
デジタル署名	ECDA P-256 曲線 [X9.62]
ハッシュ	SHA-256 [FIPS 180-4]
鍵共有	ECDH P-256 曲線 [SEC1]
鍵配付	SHA-256 に基づく [SEC1]
鍵ラップ	AES-128 [RFC 3394]
暗号化	AES-128 CBC モード [FIPS 197] および [SP 800-38A]

*[\[RFC 6318\]](#)を参照

²² これらのサービスは、S/MIME V3 規格[\[RFC 2634\]](#) で定義されている。

表 5-3 : 暗号スイート B、レベル 2*

メカニズム	ガイダンス
デジタル署名	ECDA P-384 曲線 [X9.62]
ハッシュ	SHA-384 [FIPS 180-4]
鍵共有	ECDH P-384 曲線 [SEC1]
鍵配付	SHA-384 に基づく [SEC1]
鍵ラップ	AES-256 [RFC 3394]
暗号化	AES-256 CBC モード [FIPS 197] および [SP 800-38A]

*[\[RFC 6318\]](#)を参照

連邦政府クライアントは、送信者および受信者の有効な連邦政府 PKI X.509 証明書を持った公開鍵基盤によってサポートされなければならない。

連邦政府システムで使用される暗号モジュールは、FIPS 140-2 [\[FIPS 140-2\]](#) に適合しなければならない。

連邦政府 S/MIME 実装は、組織の方針に従い、送信する保護されたメッセージでの連邦政府用途のために承認されていないアルゴリズムスイートで保護されたメッセージを受信できてもよい。それらの例において、利用者には、暗号メカニズムが弱いこと、およびそれゆえに、完全性と認証が保証されることが可能でないことを説明する警告バナーが表示されるべきである。

5.3 調達ガイダンス

以下の推奨事項は、S/MIME 利用可能な要素を調達する購入決定を行う任意の個人のためのものである。

1. 連邦政府横断的なセキュリティと互換性のサポートにおいて、すべての連邦政府情報システムが暗号スイート 1 をサポートすることを保証する。
2. 調達は、暗号スイート B、以下のいずれか、レベル 1、レベル 2、または両方をサポートするべきである。
3. 連邦政府クライアントは、より弱い暗号かつ承認されていないアルゴリズムで暗号化された通信を利用者が受信したような事象においてのみ、RC2 をサポートしてもよい。
4. 連邦政府機関がデジタル署名に SHA-1 を使用しないことを保証する；しかし、SHA-1 を用いて署名されたデジタル署名の検証に使用することはできる。暗号アルゴリズム実装は、新しいアルゴリズムを許容するためにモジュラー式であるべきである。

S/MIME クライアントが鍵ペアを生成する必要がある場合、S/MIME クライアントまたは何らかの関連する管理ユーティリティまたは管理機能が利用を代行して公開・プライベート鍵ペアを生成できることを保証する。

5.4 システムインストーラのための推奨事項

システムインストーラとは、S/MIME アプリケーションをインストールし、システムの初期設定を行う個人である。

1. 連邦クライアントは、[セクション 5.2](#) で上述のとおり相互運用性のため、暗号スイート 1 をサポートするよう構成されなければならない。
2. システムは、新しいメッセージを暗号化または署名するために、承認された暗号アルゴリズムおよび承認された鍵サイズの使用のみを許可するように構成されなければならない。
3. インストーラは、承認された暗号アルゴリズムスイートをデフォルトで使用するように S/MIME クライアントインストールし、構成しなければならない。さらに、インストーラは、相互運用性のために求められ、組織のニーズと方針に従ってエンドユーザがデフォルト設定を変更し、アルゴリズム選択のための直接的な手段があるようにクライアントを構成するべきである。
4. システムインストーラは、エンドユーザが廃棄時に各セキュリティ機能（例、暗号化、デジタル署名）の一意な証明書を使用できるようにクライアントを構成するべきである。

5.5 システム管理者のための推奨事項

システム管理者とは、日々 S/MIME アプリケーションを実行し、クライアント実装に関してエンドユーザと対話するような個人である。

1. システム管理者は、エンドユーザが適切に訓練され、組織のセキュリティ方針が実施されることを保証しなければならない。
2. システムは、新しいメッセージを暗号化または署名するために、承認された暗号アルゴリズムの使用と承認された鍵サイズの使用のみを許可するように維持されなければならない。
3. 管理者は、デフォルトで承認された暗号アルゴリズムスイートを使用するように S/MIME クライアントを維持するべきである。さらに、管理者は、相互運用性のために求められ、組織のニーズと方針に従ってエンドユーザがデフォルトの設定を変更し、アルゴリズムを選択するための直接的な手段を維持するべきである。
4. システム管理者は、様々な暗号アルゴリズムによって提供される相対的なセキュリティおよびそれらの使用に関する組織の方針に関する訓練を利用者に提供するべきである。
5. システム管理者は、証明書および鍵が保管され管理される方法についてのガイダンスをエンドユーザに提供し、エンドユーザの関連する責任を識別するべきである。

5.6 エンドユーザのための推奨事項

エンドユーザとは、システムにアクセスするためにクライアントを使用する個人である。集中管理された環境内であっても、エンドユーザは、S/MIME 実装の何らかのセキュリティ機能の重要な管理を行うことがあるかもしれない。

1. 利用者は、組織およびシステム管理者によって指示されるとおりに彼らのシステムを操作しなければならない。
2. 利用者は、廃棄時²³に各セキュリティ機能のための一意な証明書を使用するべきである。
3. 利用者は、許可されない暴露からプライベート鍵を保護しなければならない。
4. 利用者は、暗号化されたテキストと平文の両方で同じメッセージを送信するべきでない。

²³ 親元の組織によって証明書が供給されなかった場合、利用者は、ウェブ経由で多くの組織からの証明書を取得できる。

6 ケルベロス (Kerberos)

6.1 説明

ケルベロス認証メカニズムは、利用者²⁴を代行してクライアントソフトウェアが動作するような、保護されないネットワーク越しに対象サーバ (TS: Target Server) への利用者のセキュアな認証を可能とするために、マサチューセッツ工科大学 (MIT) で開発された。ケルベロスの当初の設計と実装、およびその最初の3つの改訂 (即ち、バージョン1から4まで) は、主にスティーブ・ミラー氏、クリフォードニューマン氏、ジェローム・ソルツァー氏とジェフリー・シラー氏²⁵の業績であった。ケルベロスは、ローカルログイン、リモート (ネットワーク越え) 認証、およびクライアントから TS への要求のために使用される。クライアントと TS の間の暗号鍵の確立のために提供されるようにも拡張が可能である。ケルベロスは、それぞれの当事者の識別情報の保証を提供するために利用者と TS が信頼されるサードパーティに頼るように設計されている。この保証は、それぞれ対称鍵で暗号化された、チケットと認証情報によって授与される。

信頼されるサードパーティとは、認証サーバ (AS: Authentication Server) とチケット授与サービス (TGS: Ticket Granting Service) からなる、鍵配付センター (KDC: Key Distribution Center) である。AS と TGS は、同じ機械上に存在してもよいし、しなくてもよい。KDC は、利用者、TS、および TGS 対称鍵のデータベースを持っている。すべての KDC 対称鍵は、TGS によってアクセス可能である。利用者の鍵は、通常利用者のパスワードをその他の情報と共にハッシュすることによって生成される。

ケルベロスバージョン5プロトコルの概要は、図6-1に示される。以下は、プロセスの単純化したものである (例. ほとんどの鍵の生成とほとんどの暗号操作の仕様は規定されない)。例えば、チケットと認証情報は、送信時のチェックサムと暗号化で保護される。

1. 利用者対称鍵が生成されるようなクライアントからの、パスワード入力によるクライアントへの利用者ログ
2. 利用者を代行して動作する、クライアントは、AS からのチケット授与チケットを要求する。
3. AS は、規定された有効期間についてのチケット授与チケットを生成し、それをクライアントへ送付する。
4. クライアントは、自身の認証情報と共にクライアントの識別情報とタイムスタンプを含めて、チケット授与チケットを TGS へ提供する。
5. TGS は、認証情報とチケット授与チケットの有効期間をチェックする。TGS は、次に対象サーバチケットを生成し、それをクライアントへ送付する。
6. クライアントは、認証情報と対象サーバチケットを TS へ送付する。
7. TS は、認証情報と対象サーバチケットの有効期間をチェックする ; 情報が妥当であれば、利用者は TS に対して認証される。

プロトコルは利用者に対して TS を認証するように拡張されることもあり、チケットが有効期間内に再利用されてもよい。

²⁴ 1つのクライアント実装が複数の利用者によって使用されてもよいし、1人の利用者が複数のクライアント実装を使用してもよい (例、利用者は、自身のクライアント実装をそれぞれ異なる複数のワークステーションにアクセスできる)。

²⁵ 設計は Needham と Schroeder によって提案されたプロトコル [\[NEED\]](#) と共に Denning と Sacco によって提供された修正 [\[DENN\]](#) に一部基づいていた。ケルベロスの目標、動機および根拠のさらなる詳細については [\[NEUM\]](#) を参照。

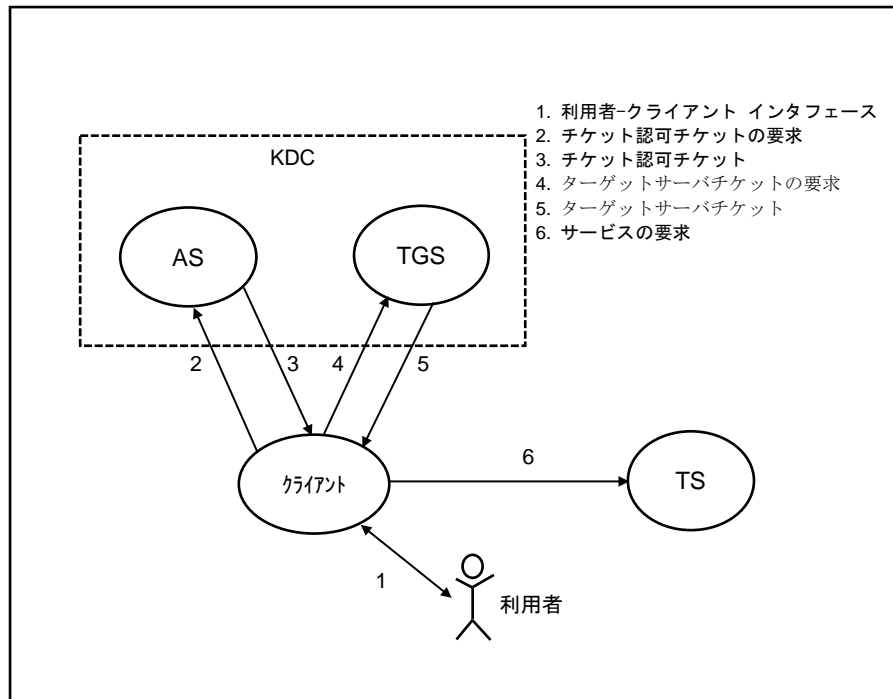


図 6-1 : ケルベロスプロトコル

各 TGS は、クライアントと TS の自身の“領域 (realm)”を持つ。しかし、異なる領域が複数の TGS 間で領域間鍵の共有によってリンクされるかもしれない (図 6-2 を参照)。領域 2 の TS 上のサービスを希望する領域 1 のクライアントは、TGS2 に対してクライアントを紹介するような、TGS1 からのチケットを入手するかもしれない。このチケットは、TGS1 と TGS2 の間で共有される領域間鍵で暗号化される。クライアントは次に領域 2 の TS 上での希望するサービスのための TGS2 からのチケットを要求することができる。従って、複数の領域は、領域間サービスを用いてクライアントに提供するため、ネットワーク化されることがある。

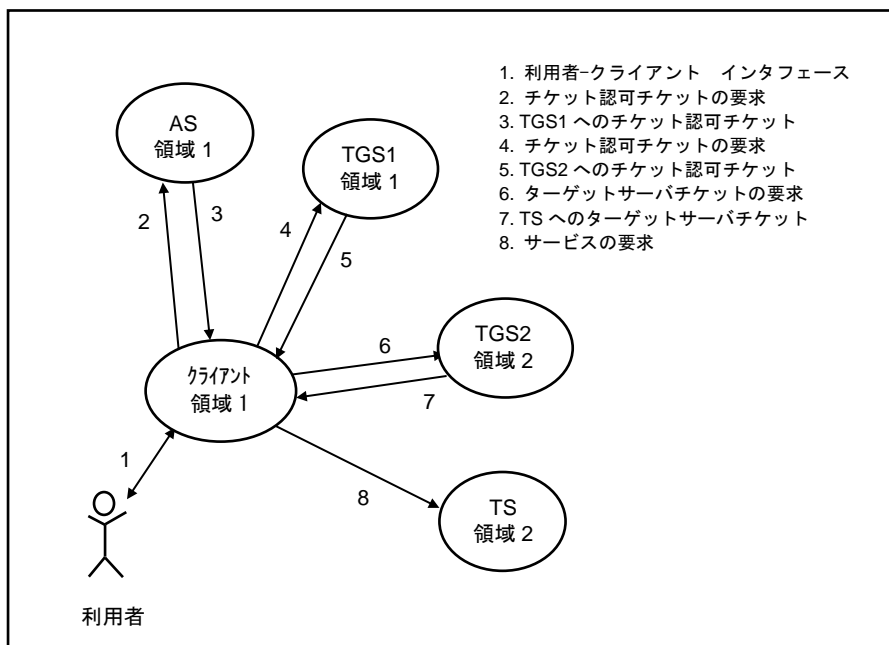


図 6-2 : 領域間認証

クライアントと AS 間の別のケルベロスプロトコル ([\[RFC 4556\]](#)で規定されるとおり) では、利用者と AS の両方が鍵確立公開鍵ペアと対応する証明書を持っているか、利用者が鍵確立用鍵ペアと対応する証明書を持ち、AS がデジタル署名用鍵ペアとデジタル署名証明書を持っているかのいずれかである。利用者対称鍵は、そのときクライアントと KDC の間で以下の 2 つのうちの 1 つの方法で確立可能である：

1. AS とクライアント間の鍵共有 (例. Diffie-Hellman) を用いる ²⁶、または
2. AS が利用者対称鍵を生成、クライアントにその鍵を送付するような、鍵配送 (例. RSA) を用いる ²⁷。

一度、利用者対称鍵が確立されると、プロトコルの残りは上述のとおりに進む。この場合、利用者対称鍵のパスワードからの生成の必要性は避けられる。

ケルベロス実装は、[\[RFC 6113\]](#) で記述されるとおり、事前認証と呼ばれるような追加の認証機能を提供してもよい。TLS が、[\[RFC 6251\]](#)で記述されるとおり、クライアントと KDC との間のすべての通信を保護するためにも実装されてもよい。

6.2 セキュリティおよび適合性の問題

1. ケルベロスのバージョン 5 は、[\[RFC 1510\]](#) で当初は規定された。さらに最近、セキュリティがバージョン 5 ([\[RFC 4120\]](#)および[\[RFC 6649\]](#)) においてアップデートされた；しかし、多くの既存の実装は、まだ初期の RFC に相当する。
2. [\[RFC 1510\]](#) に基づく、現在の多くのケルベロス実装は、対称鍵暗号機能として DES に頼っている。DES は連邦政府情報保護の用途としてもはや承認されておらず、ケルベロス[\[RFC 6649\]](#)でも同様に非推奨となっている。
3. キーレスチェックサム計算が、ケルベロスメッセージのデータ完全性のために使用される場合、メッセージの完全性は不十分でもよい。
4. いくつかのケルベロス実装は、クライアント対称鍵 (クライアント鍵は、利用者パスワードのハッシュである) を生成するために利用者パスワードによって提供されるエントロピー (例. ランダムさ) だけに依存している。一般に、パスワードは鍵を生成するための十分なランダムさを提供しない。このような場合、辞書攻撃 ²⁸が可能である。パスワードが暗号鍵を生成するために使用される場合、それらは、パスワード推測攻撃の困難さを最大化し、したがってオフラインの辞書攻撃の困難さ [\[SP 800-118\]](#) ²⁹を増大されるように、選択されるべきである。
5. クライアント、KDC または TS を危殆化することは、それらが含むような対称鍵を危殆化し、その結果、システムの一部を危殆化する。特に、KDC は、KDC TGS と直接通信するようなすべての KDC 利用者、TGS、および任意の TS の鍵情報を保管する。これらの対称鍵は、それらが保護するようなデータに要求される保護に見合った保護を要求する (例. チケット、その他の鍵、認証情報、および共有データ)。
6. TGS は、KDC データベースへの読み出しのみのアクセスを有する。TGS とデータベースが同

²⁶ この場合、利用者と AS の両方が鍵確立用鍵ペアを持つ。

²⁷ この場合、利用者は鍵確立用鍵ペアを持ち、AS はデジタル署名鍵ペアを持つ。

²⁸ 辞書攻撃は、辞書で通常見つかる言葉のリストからパスワード候補を選択することによってパスワードを推測する手法であり、または辞書で通常見つかる言葉から導出される。それぞれの選択された候補は、テスト結果が正しいパスワードが選択されることを示すまで実際のパスワードであるかのようにテストされる。

²⁹ ドラフト版 SP 800-118、エンタープライズパスワード管理へのガイド、は現在開発中である[\[SP 800-118\]](#)。

じ機械上にない場合、TGS が要求される TS 鍵を取得するためにはセキュアチャネルが要求される。

7. AS または TGS の故障は、すべての AS 利用者が新しいチケットの取得と新しいサービスの通信を阻害することになる。
8. クロックは、クロック認証情報とチケットの有効性に正確にアクセスするために同期されなければならない。TS のクロックが KDC のクロックよりも遅れている場合、以前の認証情報とチケットが期限切れとなった後でもプレイバックされることが可能となる。
9. ケルベロス実装が TLS 機能を有する場合、上述の DH 鍵共有または RSA 鍵配送方式は使用されない。

6.3 調達ガイドライン

以下の推奨事項は、ケルベロス機能を取得するための調達決定を行う任意の個人のためのものである。

1. 新たな調達がバージョン 5 ([\[RFC 4120\]](#)、[\[RFC 6649\]](#)) に適合することを保証する。
2. 政府調達が承認された対称鍵暗号アルゴリズム (例. 高度暗号化規格 (AES)) [\[RFC 3962\]](#)を含むことを規定することを保証する。
3. 承認された MAC 計算 (例. HMAC-SHA1 または HMAC-SHA256-128) が暗号化付きのデータ完全性 ([\[RFC 3962\]](#)および IETF インターネットドラフト、ケルベロス 5 の *HMAC-SHA2* を用いた AES 暗号化、<http://tools.ietf.org/html/draft-ietf-kitten-aes-cts-hmac-sha2-04>) のために利用可能であることを保証する。
4. ケルベロス バージョン 5 は、利用者のパスワードを保管するため、スマートカードやトークン (例. FIPS 201 個人識別情報検証カード [\[FIPS201\]](#)) の使用を許可する。トークン上に保管されるパスワードがランダムに生成されることを保証する；ゆえに、トークンが使用されるとき、ランダムなパスワードを生成する手段、およびトークン上にそれらをセキュアな書き込みが可能であること、およびパスワード生成機能が NIST 承認された乱数生成機能であることを保証する [\[SP 800-90A\]](#)。トークンが使用されるとき、パスワードの手動入力が入力されたトークンへの利用者の認証を除き許可されないことを保証する。
5. パスワードが利用者対称鍵を形成するために使用される場合、パスワードメカニズムが強いパスワード [\[SP 800-118\]](#) の利用をサポートすること、および承認されたハッシュアルゴリズムがハッシュアルゴリズム (例. SHA1 またはより強いもの) として使用されることを保証する。
6. パスワードが利用者によって生成される場合、システムソフトウェアが [\[SP 800-118\]](#) に従って強いパスワードポリシーを実施することを保証する。
7. 公開鍵認証とその後の鍵確立を用いるケルベロスは、パスワードベースの鍵の使用よりも強いセキュリティを提供することができる、また PKI メカニズムが利用可能な場合に利用可能であるべきである。より詳細な情報については、[\[RFC 4556\]](#)および[\[RFC 5349\]](#)を参照。使用される鍵確立方法 (即ち、鍵共有または鍵配送方法) が少なくとも 112 bits のセキュリティ強度を持つことを保証する；さらなる情報については、[\[SP 800-57 Part 1\]](#)および[\[SP 800-131A\]](#)を参照。
8. システムがサポートする場合、TLS が使用されるべきである。
9. 調達事務官は、領域間ネットワークが必要かどうか、また必要に応じてソフトウェアにおいて機能を含めるかどうかをについて検討するべきである。
10. CA、TS およびクライアントによって使用される暗号モジュールが FIPS 140-2 レベル 1 またはそれ以上 [\[FIPS 140-2\]](#) で認証されていることを保証する。

6.4 システムインストーラのための推奨事項

システムインストーラとは、ケルベロス機能をインストールし、システムの初期設定を行う任意の個人である。

1. 政府システムは、承認されたアルゴリズム（例、AES）が使用されなければならない[RFC 3962]、かつ DES が使用されてはならない[RFC 6649]ように構成されなければならない。
2. 承認された MAC チェックサム（例、HMAC-SHA1 または HMAC-SHA256-128）が、データ完全性目的のすべての実装で、インストールされ、使用されなければならない（[RFC 3962]および <http://tools.ietf.org/html/draft-ietf-kitten-aes-cts-hmac-sha2-04>）にて、IETF Internet Draft, *AES Encryption with HMACOSHA2 for Kerberos 5* を参照）。
3. AS、TGS、TS、およびクライアントは、鍵の保護と更新のために、強いアクセス制御メカニズム³⁰（物理的および論理的な）を使用しなければならない。
4. ケルベロス バージョン 5 は、利用者のパスワードを保管するため、スマートカードまたはトークン（例、FIPS 201 個人識別検証カード[FIPS 201]）の使用を許可する。トークンに保管されるパスワードは、ランダムに生成されなければならない；ゆえに、トークンが使用される場合、NIST 承認された乱数生成機能（[SP800-90A], [SP800-90B], [SP800-90C]）によってランダムなパスワードを生成する手段、およびセキュアにそれらをトークンに書き込む手段が使用されなければならない。トークンが使用される場合、手動によるパスワード入力、トークンに対して利用者を認証する以外に許可されてはならない。
5. 公開鍵ベースの利用者認証と鍵確立を伴うケルベロスは、パスワードベースの鍵よりも強いセキュリティを提供でき、また PKI メカニズムが利用可能でソフトウェアがその機能を持っている場合、インストールされるべきである。さらなる情報について、[RFC 4556]を参照。
6. システムが TLS をサポートする場合、TLS が使用されるべきである、TLS の使用にかするガイドラインについては、[セクション 4](#)を参照。
7. 利用者パスワードがシステムによって生成される場合、システムは、強いパスワードを生成しなければならない[SP 800-118]。パスワードが利用者対称鍵を形成するために使用される場合、承認されたハッシュアルゴリズム（例、SHA1 またはより強いもの）がパスワードハッシュアルゴリズムとして使用されなければならない。
8. 利用者パスワードが暗号鍵を生成するために使用される場合、パスワードメカニズムが使用されるよう構成され、強いパスワードを要求しなければならない[SP 800-118]。
9. バックアップ AS と TGS は、運用上の障害または DOS 攻撃の場合に影響を最小化するよう提供されるべきである。
10. クロックは、定期的に同期され、いつでも新しいシステムがオンライン接続されている³¹べきである。

³⁰ 強いアクセス制御メカニズムは、機微なデータをアクセスまたは置換するような許可されない試みを防止又は関知のいずれかを行う。これらの制御は、物理的（例、ロック、ガード、またはアラーム）または論理的（例、暗号化、データ完全性、またはエンティティ認証）であってもよい。

³¹ <http://www.nist.time.gov> を参照。

6.5 システム管理者のための推奨事項

システム管理者とは、ケルベロス機能を持つシステムを日々実行させ、エンドユーザと対話する任意の個人である。

1. システム管理者は、利用者が適切に訓練を受けること、および組織のセキュリティ方針が実施されることを保証しなければならない。
2. AS、TGS、TS、およびクライアントが物理的にセキュアとされるようにしなければならない。
3. チケットは、クライアント、TS、および TGS サイトにおいて暗号化され、または物理的に保護されなければならない。
4. パスワードが利用者によって生成される場合、システム管理者は、ソフトウェアによって実施される強いパスワードの選択のための方針を作らなければならない[[SP 800-118](#)]。
5. システムクロックは、同期を保証するため、定期的に検証されなければならない。

6.6 エンドユーザのための推奨事項

エンドユーザとは、ケルベロス機能を使用する個人である。

1. 利用者選択のパスワードが許容される場合、それらは、組織のパスワード方針に従って生成されなければならない。
2. 利用者は、許可されない暴露からそれらのパスワードを保護しなければならない。パスワードまたは鍵を含むトークンが提供される場合、利用者は、許可されない使用からトークンを保護しなければならない。利用者は、物理的なトークンの喪失またはパスワードの危殆化を報告しなければならない。

7 無線回線経由の鍵更新 (OTAR) 鍵管理メッセージ (KMM)

7.1 説明

デジタル無線の無線回線経由の鍵更新 (OTAR) [\[OTAR\]](#) のために鍵管理プロトコルが規定されている。このプロトコルは、暗号的セキュリティのいくつかのタイプを取り扱うために設計されている、そのうちの1つ、タイプ3は、非格付け (訳注：機密でないもの) の、機微な通信であり、ここで議論されているもの、のために設計されている。複数のセキュリティタイプ間の唯一の違いは、使用される暗号アルゴリズムとセキュリティ要求事項である。タイプ3アルゴリズムとセキュリティ要求事項は、OTAR と OTAR1 [\[OTAR1\]](#)³²の両方において対処される。

鍵管理について、セキュアモバイルシステムは、鍵管理ファシリティ (KMFs : Key Management Facilities) と各 KMF の配下のモバイルラジオからなる。鍵管理メッセージ (KMMs : Key Management Messages) は、各 KMF とその配下のモバイルラジオの間で交換される (図 7-1)。暗号鍵は、KMF からモバイルラジオへ送信され、鍵ラッピングアルゴリズムと鍵ラッピング用鍵を用いて保護される；多くの KMM がメッセージのデータを暗号化することによって保護される；メッセージの完全性はメッセージ認証コード (MAC) を用いて保護される。

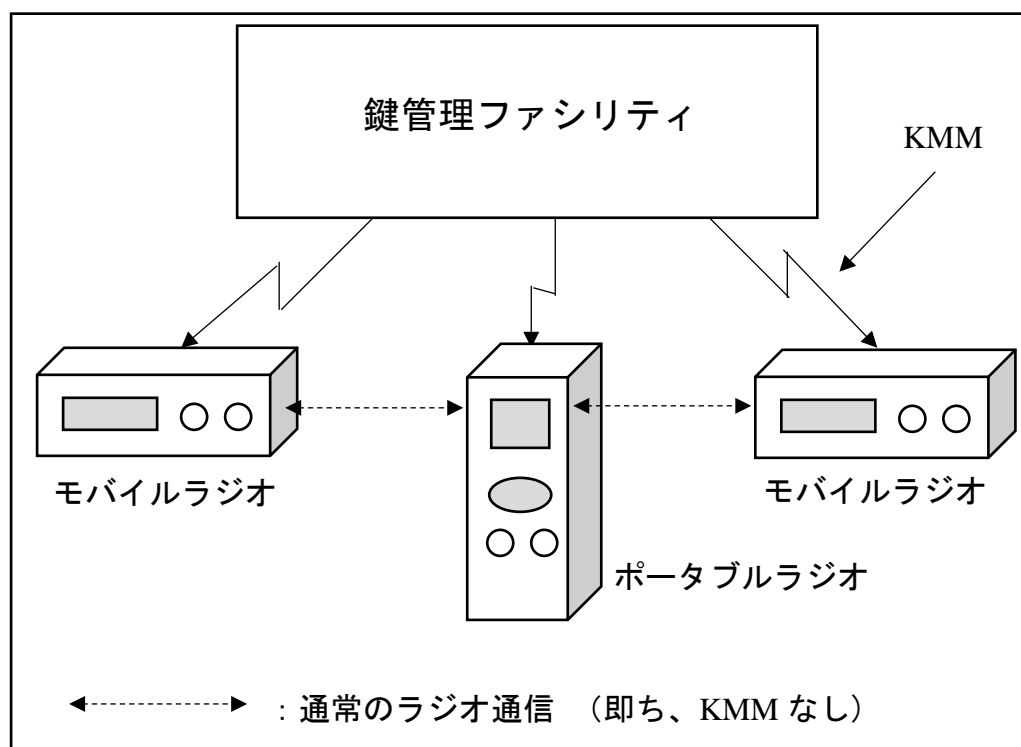


図 7-1 : OTAR 搭載のラジオ通信

3 つの一般的な鍵タイプが OTAR では使用される：鍵ラッピング用鍵 (KWK : Key-Wrapping Key)³³、トラフィック暗号化鍵 (TEK) およびメッセージ認証コード (MAC) の計算のために使用される鍵。

³² 参照 [\[OTAR\]](#) は、鍵管理手法とプロトコルの概要を提供する。参照 [\[OTAR1\]](#) は、タイプ3 鍵管理メッセージ (KMM) を送信するための一般的なセキュリティ要求事項、鍵をラッピングするための要求事項、KMM 完全性のために使用される手法および KMM のリプレイに対する保護のために使用されるメカニズムを規定する。

³³ 鍵ラッピング用鍵は、鍵暗号化用鍵 (KEK) としても参照される。

7.2 セキュリティおよび適合性の問題

7.2.1 暗号アルゴリズム

プロトコルは、暗号保護を適用する任意のブロック暗号アルゴリズムの使用を許可するよう設計されているが、3つのブロック暗号アルゴリズムのみが使用に含まれている：DES、TDEA および AES。

連邦政府情報を保護する必要があるセキュリティをもはや DES は提供しないため、DES の承認は取り消された。

[\[SP 800-67\]](#) で規定されるとおり、TDEA は、3つの別々の DES 鍵からなる“鍵バンドル”を持つ3回の DES 暗号化/復号操作を使用する。TDEA の2つのバージョンが OTAR 使用で含まれている：3つの鍵すべてが同じで DES と互換性のある、ワンキー（1つの鍵）バージョン、および3つの鍵が異なる、スリーキー（3つの鍵）バージョン（TDEA）。DES がもはやセキュアであると思われなければならないため、TDEA のワンキーバージョンについてももはやセキュアであると思われなければならない。

7.2.2 メッセージ認証と暗号周期

メッセージ認証コード（MAC）は、OTAR1 で規定されるとおり、CBC-MAC モードの操作を用いて、多くの KMM を認証し、その完全性を保護するために使用される。MAC のセキュリティは、一部において、MAC アルゴリズムのブロック長に依存する。AES は3-TDEA よりも大きいブロック長を持つので、AES CBC-MAC のセキュリティは、3-TDEA CBC-MAC よりもより良い。OTAR 文書は暗号周期の長さに関して一切ガイダンスを提供していない（即ち、メッセージ数または鍵が変更されなければいけなくなる前に使用されてもよい時間の長さ）。

AES について、所与の鍵を用いて認証可能なメッセージの数は、実際には、問題ではない。しかし、AES 鍵は、システムに対するその他の脅威、例えば、ラジオの紛失、または鍵の検知されない危殆化等の理由で、定期的に更新されなければならない。

3-TDEA を用いるとき、1 000 000 以上のメッセージが所与の鍵を用いて送信されてはならない、なぜなら、アルゴリズムのセキュリティに対する脅威のため。しかし、システムに対するその他の脅威のため、AES のように、3-TDEA 鍵をより頻繁に更新することは、賢明である。

7.2.3 鍵の用途

本推奨事項の第一部は、鍵は1つの目的だけ³⁴のために使用されなければならないと述べている。しかし、OTAR1 は、MAC を生成するために使用される鍵は、認証及び完全性保護目的のために予約された鍵、または鍵ラッピングアルゴリズムを用いたトラフィック暗号化鍵（TEK：Traffic-Encryption Key）から導出される鍵のいずれかでなければならないと述べている。後者の場合、TEK は、暗号化と鍵導出の両方のために使用されるかもしれないことに注意すること。1つの目的だけのためにある鍵を使用するという推奨事項に適合するため、MAC 鍵は、1つの目的のために予約された鍵でなければならない。

7.2.4 バックアップ

KMF は、モバイルラジオと共有するすべての鍵材料を、必要に応じて回復可能となるように、バックアップすべきである。ある鍵がもはや要求されないとき、それは通常環境の記憶及びバックアップ記憶の両方から削除されるべきである。

³⁴ この規則に対して許可された例外があるが、OTAR には適用されない[\[SP 800-57 Part1\]](#)、セクション 5.2]。

7.2.5 鍵更新

手順は、鍵危殆化の事象において、ネットワーク内のすべてのラジオの鍵更新を行うべきである。ラジオを紛失した場合、手順は、紛失したラジオがもはやネットワーク内のその他のラジオとセキュアに通信する能力を持たないようにネットワーク内のその他のラジオの鍵更新を有効化しなければならない。

7.2.6 乱数ビット生成器

鍵は、暗号処理の望まれるセキュリティ強度に対して十分なランダムさを提供するような承認された乱数ビット生成器を用いた KMF において生成されなければならない。NIST 承認された乱数ビット生成方法が [\[SP 800-90A\]](#)、[\[SP 800-90B\]](#) および [\[SP 800-90C\]](#)（開発中）において規定される。

7.3 調達ガイダンス

以下の推奨事項は、OTAR 装置取得の調達する決定をおこなうような任意の個人のためのものである。

1. AES または TDEA アルゴリズムが含まれていることを保証する。
2. TDEA が実装において提供される場合、スリーキー（3 つの鍵）バージョンが含まれ、実装は 1 つの TDEA 鍵の使用制限数が 1 000 000³⁵ にバンドル可能であることを保証する。
3. KMF とラジオが OTAR と OTAR1 に適合することを保証する。
4. 鍵が KMF ないで生成されるとき、それらが承認された乱数ビット生成器を用いて生成されることを保証する。
5. KMF およびモバイルラジオによって使用される暗号モジュールが FIPS 140-2 レベル 1 またはそれ以上 [\[FIPS 140-2\]](#) で認証されていることを保証する。
6. KMF が災害事象（例. 火災、地震等）において KMF の再構築をサポートするためのバックアップおよびアーカイブ機能を含むことを保証する。

7.4 システムインストーラのための推奨事項

システムインストーラとは、OTAR 機能をインストールし、システム要素の初期設定を行うような個人である。

1. KMF は、暗号鍵を保護するため、強い物理的および論理的アクセス制御メカニズムを持つとともに使用しなければならない。
2. バックアップ KMF は、提供されなければならない。
3. TEK は、複数の目的で使用されてはならない。予約された MAC 鍵は、メッセージ認証及び完全性保護のために使用されるべきである。
4. それぞれの鍵タイプの最大暗号周期は、組織のセキュリティ方針と [セクション 7.2.2](#) に従って、KMF において決定されなければならない。
5. ラジオは、以下についての説明責任を負う；ラジオが紛失または盗難に合った場合、ラジオに含まれる鍵の喪失の影響の評定がなされなければならない。そのラジオに含まれる任意の鍵の使用は、中止されなければならない。手順は、KMF またはその他のラジオによって使用される場合、これらの鍵を置換するようにされなければならない。

³⁵ [セクション 7.2.2](#) を参照。

6. 実装は、AES または TDEA アルゴリズムを使用し、DES の使用を拒否するよう設定されなければならない。
7. TDEA が提供され、使用される場合、スリーキー（3 つの鍵）バージョンが使用されなければならない、かつワンキー（1 つの鍵）バージョンが使用されてはならない。
8. 鍵バンドルの暗号周期が設定可能な TDEA を用いた実装について、暗号周期は 1 000 000 メッセージよりも少ない値に設定されなければならない。

7.5 システム管理者のための推奨事項

システム管理者とは OTAR システムまたはその要素を日々管理し、エンドユーザと対話するような個人である。

1. システム管理者は、組織のセキュリティ方針が実施されることを保証しなければならない。
2. システム管理者は、鍵材料を暴露や改ざんから保護しなければならない。
3. 手順は、鍵の暗号周期の終了時に鍵の置換を行わなければならない。
4. KMF の可用性を維持するため、システム管理者は、災害の後、KMF を再構築するためのセキュアなロケーションに十分な情報が保管されることを保証しなければならない。
5. 戦略に沿ったバックアップ KMF および当初の KMF からバックアップ KMF へ、およびバックアップ KMF から当初の KMF への移行手順が確立されなければならない。
6. システム管理者は、ラジオの使用法、およびラジオを紛失または鍵が危殆化した場合に従うべき手順についてエンドユーザを訓練しなければならない。
7. 監査ログは、KMF において、どの鍵がどのラジオにより共有されているかを示す十分な情報と共に維持されるべきである。

7.6 エンドユーザのための推奨事項

エンドユーザとは、OTAR 機能を持つラジオを用いる個人である。

1. エンドユーザは、組織およびシステムの管理者による指示のとおり、ラジオを操作しなければならない。
2. エンドユーザは、紛失および許可されないアクセスからラジオを保護しなければならない。
3. ラジオが紛失したり、または鍵が危殆化したりと疑われるような事象において、エンドユーザは、組織のセキュリティ方針に従ってシステム管理者に直ちに通知しなければならない。

8 ドメインネームシステム セキュリティ拡張 (DNSSEC)

8.1 説明

ドメインネームシステム (DNS) は、[\[RFC 1034\]](#)と[\[RFC1035\]](#)で定義されるとおり、インターネットアドレス、シンプルメール転送プロトコル (SMTP) サーバ、およびその他の情報を、人が判読可能な名前にマッピングするためのグローバルな階層構造の分散型データベースシステムである。主な目的はホストドメイン名とインターネットアドレス間のマッピングを取り扱うことであるが、その他の形のデータについても同様に、例えば、ホストシステム情報、サーバの地理的ロケーション、エンコードされたデジタル証明書でさえも取り扱うことができる。DNS データは、1 片のデータをドメイン名にそれぞれ結び付ける個別のリソースレコード (RR)、およびリソースレコードを識別するタイプコード (RR type) として保管される。特定の組織のすべての RR は、ゾーンと呼ばれる管理ユニットで保管される。複数のゾーンがドメインを形成する。ドメインは、あるゾーンが一つまたは複数の子-委任ゾーンへ委任する親としてふるまうような階層構造を持つ。例えば、ほとんどの連邦政府機関は、“.gov” 親ゾーンの下で子としての委任されたものである。

ゾーン情報は、信頼すべきサーバ上で維持され、DNS ネットワークプロトコルに従った問合せに回答するためにインターネット全体に配付される。DNS 基盤は、ローカルのゾーンデータベースを持つプライマリマスター権威サーバ、および、プライマリ権威マスターサーバからのゾーンデータベースのコピーを取得するような複数のセカンダリサーバとして知られる、小さいグループ (または1つのサーバ) からなる。別の要素のセットは、権威サーバに問合せでキャッシュが回答するような、キャッシュ再帰サーバ³⁶である。エンドユーザのクライアントシステム上で、リゾルバとして知られるソフトウェア要素は、再帰キャッシュおよび/または権威サーバに対して DNS 問合せを行う。図 8-1 は、DNS 要素間の関係を図示する。

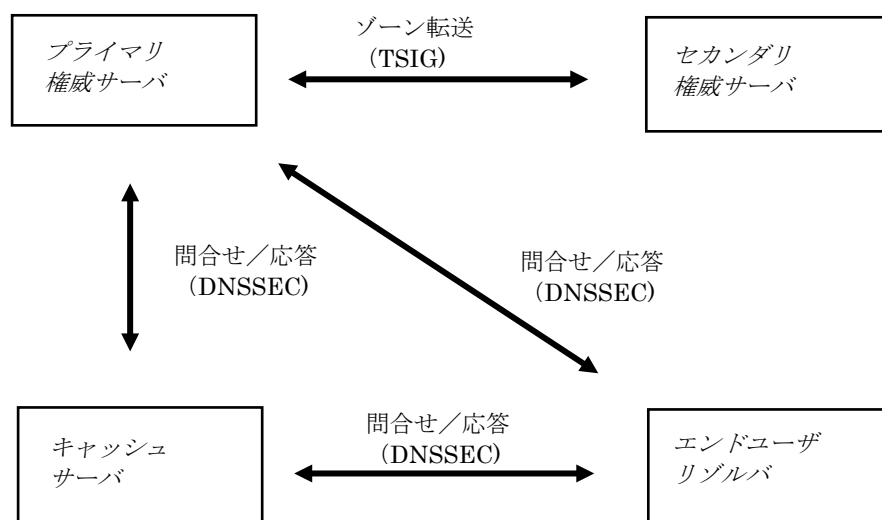


図 8-1 : DNS 要素

基本の DNS は、多くのセキュリティ機能を持っていない[\[SP 800-81\]](#)。3つの IETF 文書に含まれるような、総称として DNS セキュリティ拡張 (DNSSEC) ([\[RFC 4033\]](#)、[\[RFC 4034\]](#)、[\[RFC 4035\]](#)) と呼ばれる

³⁶ キャッシュ再帰サーバは、“キャッシュサーバ” または “再帰サーバ” と短縮して呼ばれることがときどきある。しかし、役割は同じである。

セキュリティ強化を提供するため、RFC スイートが開発されてきた。DNSSEC は、1 階層の認証とその他のプロトコルで使用されるデータを含めて、DNS で保管されるあらゆる種類のデータの完全性保護を提供する。例えば、DNS でのセキュアシェル (SSH) 鍵を保管するために割り当てられる RR タイプがあり、その情報の完全性を保護するために DNSSEC を信頼する。

8.1.1 DNS データ認証

暗号的に生成された公開鍵ベースのデジタル署名は DNS データのための認証を提供する。通常、あるゾーンには、DNSSEC を実装するために使用される 2 つ以上のデジタル署名公開鍵ペア (鍵セットを構成するようなもの) がある。1 つの鍵ペアは、ゾーンデータに署名するために使用され (ゾーン署名鍵 (Zone Signing Key) または ZSK として参照される)、別の鍵ペアは、ゾーン鍵セットに署名するために使用される (鍵署名鍵 (Key Signing Key) または KSK として知られる)。いくつかもゾーンが ZSK と KSK の両方の 1 つの鍵ペアのみを使用することができるが、連邦政府機関のゾーンとしては推奨されない。この KSK は、ゾーンのセキュアエントリーポイント (SEP:Secure Entry Point) としても知られており、それを用いて、クライアントは ZSK を認証し (KSK 公開鍵を用いて ZSK 越しに署名を検証することによって)、次にゾーンデータを認証するために ZSK を使用する。KSK は、そのゾーンから委任している親へセキュリティチェイン³⁷をリンクするためにも使用される。KSK は、そのゾーン (例. “example.gov”) から委任している親ゾーン (例. “.gov”) [\[RFC4035\]](#)へのセキュリティをリンクするために使用され、しばしばより長い期間生きており、たまに使用される (ゾーン鍵セットに署名するためのみに使用される)。複数のデジタル署名アルゴリズムをサポート可能で、複数の鍵 (各アルゴリズムに 1 つ) があり、DNSSEC においてアルゴリズムのネゴシエーションが無いように、クライアントは特定のデジタル署名アルゴリズムを理解のみすればよい。IETF で定義されるとおり、実装が必須のアルゴリズムが 1 つあり、すべてのサーバとクライアントが理解するような合意に基づくデジタル署名アルゴリズムが少なくとも 1 つはある。

現時点では、SHA-1 と SHA-256 を用いる RSA の両方が規定されており、DNSSEC ゾーンと共に使用が可能である。ほとんどの最近の実装は両方をサポートするか、アップグレード後にそうなるかのいずれかである。初めて DNSSEC を配備するゾーンは、SHA-256 を用いる RSA によって開始することができる。SHA-1 を用いる RSA と共に初期に配備したゾーンは、SHA-256 を用いる RSA (2048-bit RSA 鍵) へ移行するべきである ; 異なる鍵サイズとハッシュアルゴリズムを伴う RSA を用いるためのガイドラインについての情報は、[\[SP 800-131A\]](#)を参照。しかし、両方のハッシュアルゴリズム (即ち、SHA-1 と SHA-256) は、SHA-256 を伴う RSA を検証できないようなクライアントシステムがまだ DNS データを認証できることを保証するため、一定の期間、DNS データのデジタル署名を生成するために使用されるべきである。この移行期間の長さは、SHA-256 を伴う RSA を理解するようなクライアントシステムソフトウェアの幅広い利用可能性と配備に依存する。

8.1.2 DNS トランザクション認証

追加の認証メカニズムは、サーバ・サーバ通信および管理的な制御のために使用される。トランザクション認証は、全 DNS メッセージとトランザクションにおいて両方の権威 DNS サーバによって知られるような秘密のランダム文字列、および元のメッセージにトランザクション署名 (TSIG : Transaction Signature) RR を追加した結果を送信することによって、HMAC を計算することによって行われる。トランザクション認証は、通常、ゾーン転送または動的アップデートのような、特別なトランザクションのために使用される。動的アップデートは、特別にフォーマットされたメッセージを送信することによって、許可された管理者に DNS データを追加または削除を許可するような機能である。これは、動的ホスト設定プロトコル (DHCP : Dynamic Host Configuration Protocol) が IP アドレスを動的に割り当てるた

³⁷ セキュリティチェイン (“認証のチェイン” としても参照される) は、その検証されるべきデータからクライアント上の信頼されたインストールされた公開鍵へとトレースバックするために使用可能なデジタル署名と公開鍵の集まりである [\[SP 800-81\]](#)。この公開鍵と署名のチェインは、PKI 証明書チェイン ([セクション 2.1](#) を参照) に類似しているが、全体が DNS 内に含まれている。

めに使用されるようなローカルエリアネットワークにおいて、頻繁に使用される。DHCP サーバは、ネットワーク変更を反映するために動的アップデートメッセージを送信することによって、DNS サーバを更新してもよい。

TSIG 認証で使用される、現時点で定義されたアルゴリズムは、SHA-1 と SHA-2 ファミリのハッシュアルゴリズム (SHA-224、SHA-256、SHA-384 と SHA-512) を用いる HMAC である。SHA-1 の仕様は、適切なランダムな秘密の文字列を伴う HMAC を用いる場合、今のセキュリティ実践としては受け入れ可能である。トランザクションに関与するすべての DNS サーバ管理者は、トランザクション認証のために使用されるアルゴリズムと秘密の文字列長について合意しなければならない、またすべての当事者が同じ秘密のランダムな文字列を持っていることを保証しなければならない (鍵を配付するための out-of-band (訳注: そのネットワークを使用しない等の別の経路による) のトランザクションを含むかもしれない)。

8.1.3 DNS 暗号アルゴリズム/スキーム、モードおよび組合せ

DNS は、複数のアルゴリズムを分離してサポートしないが、アルゴリズムとスキームのスイートを規定する。署名とメッセージ認証するゾーンデータのためのアルゴリズム/スキームの組合せは、表 8-1 ([\[RFC 6944\]](#)、[\[RFC 6605\]](#)) および表 8-2 ([\[RFC 3645\]](#)、[\[RFC 4635\]](#)) で提供される:

表 8-1: ゾーンデータ署名の推奨されるアルゴリズムとスキームの組合せ

スイート	認証	ダイジェスト	IETF 状態	連邦政府用途 のための承認 ³⁸
RSA_SHA-256	RSA	SHA-256	実装を推奨	YES
RSA_SHA-512	RSA	SHA-512	実装を推奨	YES
ECDSAP256SHA256	ECDSA	SHA-256	実装を推奨	YES
ECDSAP384SHA384	ECDSA	SHA384	実装を推奨	YES

³⁸ 承認された鍵長とアルゴリズム寿命については本ガイドの第一部を参照[\[SP 800-57 Part 1\]](#)。

表 8-2：推奨されるメッセージ認証アルゴリズム

スイート	IETF 状態	連邦政府用途のための承認
HMAC_SHA1	必須	YES
HMAC_SHA-224	オプション	YES
HMAC_SHA-256	必須	YES
HMAC_SHA-384	オプション	YES
HMAC_SHA-512	オプション	YES
GSS_TSIG ³⁹	オプション	YES

HMAC-MD5.SIG-ALG.REG.INT が広く実装されているスイートであり、しばしばデフォルト選択として設定されていることに注意するべきである。しかし、連邦政府の実装のために使用されてはならない。TSIG メッセージ認証が既存の信頼関係があるようなサーバ間で使用されるので、管理者は使用される方法と TSIG 方法と共に使用される秘密の（ランダムな）文字列について合意しなければならない。

メッセージ長の制約のため（以下の[セクション 8.1.4](#)を参照）、大きい RSA 鍵は、DNS 失敗としてクライアントによってしばしば解釈されるような、DNS トランザクション失敗の結果となるかもしれない。その代わりに、ECDSA [\[RFC 6605\]](#)のような、同じセキュリティ強度を持つが、より小さい鍵サイズを持つデジタル署名アルゴリズムが推奨される。2015 年 10 月 1 日までにゾーン署名用に ECDSA へ移行するための DNS 管理者計画、または DNS ソフトウェア要素において利用可能となり次第より早期に移行する計画が推奨される。

8.1.4 鍵サイズについての特別な検討

RSA DNSSEC 署名鍵の長さを選択するとき、必要とされる特別な検討がいくつかある。大きい RSA 鍵が、標準 UDP パケットに合わせるための応答メッセージが長すぎるといような、プロトコル問題を起こす可能性を示している。DNSSEC は、4 KB までのより大きい DNS パケット長の仕様を要求するが、実際の限界は、おおよそ 1 500 バイトか、それ以下である。

DNS 管理者は、2015 年 10 月 1 日まで、またはルータ、キャッシュおよびその他のネットワーク中間ボックスの大多数が 1 500 バイト以上のパケット長を取り扱えることが証明される（2015 年以前に）まで、1024-bit RSA/SHA-1 および/または RSA/SHA-256 の ZSK を維持することが推奨される。しかし、1024-bit RSA 鍵は、1500 バイト以上の長さの UDP パケットを取り扱うことができないような古いネットワーク要素であるかもしれない DNS クライアントの古いバージョンを使用できるようにするため、2015 年まで許容される。これは、[\[SP 800-131A\]](#)で提供されるセキュリティガイダンスに対する例外である。しかし RSA 鍵サイズの例外は鍵署名用鍵には適用されない。KSK は、本推奨事項の第一

³⁹ 秘密の鍵トランザクション認証のための包括的なセキュリティサービスアルゴリズム（GSS-TSIG [\[RFC3645\]](#)）は、いくつかのサーバ実装で見つかるかもしれない。

部[\[SP 800-57 Part 1\]](#)で定められたガイダンスに従わなければならない。

ゾーン管理者は、2015年までに、または DNSSEC 要素で ECDSA のサポートがみられるときまで、いずれかのうち早い方までに、DNSSEC ゾーン署名アルゴリズムを ECDSA へ移行することが推奨される。

DNSSEC と共に 1024-bit RSA ZSK を用いる時のリスクを最小化するため、ZSK は、より頻繁に変更されるべきである：3 か月毎に 1 回、5 から 7 日の署名有効期間と共に。[\[SP 800-81\]](#)で議論される ZSK ロールオーバーシーケンスは、DNS データにおける有効な認証のチェーンを維持することが推奨される。

8.1.5 NSEC3 のための特別な検討

情報漏えいのリスクを最小化するような、Hashed Next Secure (NSEC3) RR [\[RFC 5155\]](#)として知られる DNSSEC への特別な変形がある。DNSSEC において、クライアントは、ゾーンの内容をエラーメッセージでみつかる Next Secure (NSEC) RR タイプの一連の問合せを送信することによってマッピングすることができる。これらの NSEC RR は、問合せされた名前が存在しなかったが、その証明の一部としてゾーン内に存在する 2 つの名前についても提供するような、署名された証明を提供する。NSEC3 は、2 つの既存の名称のハッシュ値（現時点では SHA-1 のみを用いる）を用いることによりゾーン名の情報漏えいを最小化するために試行する。しかし、これは、サーバとクライアントが実行時に複数の SHA-1 ハッシュ計算を実行することができることを要求する：この方法は、複数の要求が為された時にサーバに対して Denial of Service attack（訳注：DoS 攻撃）を仕掛けるために使用することもできる。

NSEC3 は、DNS ゾーンにおけるネットワークリソースの完全なマッピングへ導く可能性のあるような、特定のクラスの情報漏えいを解決するために設計された。NSEC3 配備リスクは、最優先で必要とされるものとして、ゾーン内容の保護（例としては、DNS に含まれているかもしれない情報を個人的に識別することを保護することを含む）を越えて NSEC3 を配備することなしに、NSEC3 を使用することによってもたらされる有用性よりもしばしば大きくなる。しかし、NSEC3 対応クライアントソフトウェアを使用することは良い考えである、なぜならクライアントは DNSSEC と共に NSEC3 RR を使用するようなゾーンにアクセスするかもしれないからである。

システムインストーラと管理者は、SHA-1 から SHA-256 への移行し、SHA-256 が主なソフトウェア配備において利用可能となる時にそれを実行するための移行計画を策定するべきである。これは、SHA-256 対応の実装がインターネットコミュニティにおいて広く利用されるようになったと確認されるまでに、SHA-1 と SHA-256 ベースの NSEC3 RR の両方を配備することを含むだろう。

8.2 セキュリティ／適合性の問題

1. DNS クライアントのより古いバージョンを受け入れるために 2015 年 10 月 1 日まで 1024-bit RSA 鍵が許容されるとしても、2048-bit RSA 鍵の仕様が強く推奨される。
2. 仕様上強く必要とはされないが、鍵署名用鍵は親ゾーン（例. .gov）からそのゾーン（例. nist.gov）へのセキュリティチェーンを維持するために使用されるべきである。この KSK は、親ゾーンによって確立された方針と手順に従って、委任する親へセキュアに送信されるべきである。

TSIG 共有された秘密の文字列は、DNS メッセージトランザクションの完全性保護を提供する用途のためにランダムであるべきであり、適切なセキュリティ強度で生成されるべきである。TSIG (共有された) 秘密の文字列を用いるシステムインストーラは、どの TSIG アルゴリズムを使用するかについて合意しなければならない。

8.3 調達ガイダンス

以下の推奨事項は、ネットワーク基盤のための DNSSEC 対応の要素の取得について調達の決定を行う任意の個人のためのものである。

1. DNSSEC ユーティリティが FIPS 140-2 認証済の暗号モジュールを使用することを保証する [\[FIPS 140-2\]](#)。
2. DNS サーバソフトウェアは、ゾーン方針によって要求される場合、NSEC3 RR を生成し、提供するべきである。
3. 第一部のハッシュアルゴリズムセキュリティ強度推奨事項と一貫している HMAC を用いた TSIG メッセージ認証と共に使用するランダムな文字列を生成するために、DNS サーバソフトウェアが承認された乱数ビット生成器を使用する。
4. 利用可能な場合、セキュリティ方針によって要求されるとおり、別途ワークアプリケーションの DNSSEC 対応バージョンが調達されることを保証する。
5. HSA-256 を実装する DNSSEC ソフトウェアは利用可能なとき調達に含まれることを保証する。

8.4 システムインストーラのための推奨事項

システムインストーラとは、DNS 要素をインストールし、その要素の初期設定を行うような個人である。

8.4.1 システムインストーラのための推奨事項（権威サーバ）

1. 権威サーバのインストーラは、DNSSEC 署名されたゾーンデータを提供するために DNS 権威サーバを構成しなければならない。
2. 権威サーバインストーラは、トランザクションにおいて TSIG と共に使用するための初期のランダムな秘密の文字列を作成し構成するため、承認された乱数ビット生成器 ([\[ISP 800-90A\]](#)で議論されるとおり) を使用しなければならない。
3. 権威サーバは、第一部で規定されるとおり、デジタル署名のための推奨される鍵サイズと一貫する鍵ペアと共にゾーンデータを生成し署名するために構成されなければならない。
4. 権威サーバは、第一部で規定されるとおり、デジタル署名のための推奨される鍵サイズと一貫する鍵ペアと共に鍵セットを生成し署名するために構成されなければならない。
5. 権威サーバは、プライマリとセカンダリサーバ間のゾーン転送メッセージ認証 (TSIG 経由) のためのランダムな秘密の文字列を生成し使用するために構成されなければならない。乱数ビット生成器の処理のセキュリティ強度は、サーバによって要求されるセキュリティ強度をサポートしなければならない。

8.4.2 システムインストーラのための推奨事項（キャッシュ再帰サーバ）

1. 再帰キャッシュサーバのインストーラは、DNS サーバが DNSSEC 対応となるように構成しなければならない。
2. 再帰キャッシュサーバのインストーラは、DNSSEC 検証のために使用される少なくとも 1 つの公開鍵をインストールしなければならない。

8.4.3 システムインストーラのための推奨事項（クライアントシステム）

1. クライアントシステムは、DNSSEC 対応キャッシュ再帰サーバへ DNS クエリを送信するよう構成されなければならない。
2. クライアントシステムは、DNSSEC 対応アプリケーションが利用可能な場合、それらを使用するよう構成されるべきである。

8.5 システム管理者のための推奨事項

システム管理者とは、DNS アプリケーションを日々の業務として実行させ、エンドユーザと対話するような個人である。

8.5.1 システム管理者のための推奨事項（権威サーバ）

1. 権威サーバに関する組織のセキュリティ方針が実施されなければならない。
2. 暗号鍵は第一部で規定されるとおり保護されなければならない。
3. システム管理者は、ゾーンデータ署名リソースレコードの有効期限の終了日以前にそれらを置換しなければならない。
4. ゾーンデータ署名リソースレコードは、公開鍵に関連するプライベート鍵が危殆化した場合、DNS ゾーンの管理者が組織を離れたとき、および組織のセキュリティ方針に列挙されたその他の理由により、置換されなければならない。
5. 管理者は、プライベート鍵を取り扱い、保護するための方法を活用しなければならない（例、適切な利用者認証を要求するようなスマートカードを用いて）。
6. 管理者は、NIST Special Publication 800-81、*Secure Domain Name System (DNS) Deployment Guide* [\[SP 800-81\]](#) にて見られる鍵ライフサイクル手順に従わなければならない。

8.5.2 システム管理者のための推奨事項（キャッシュ再帰サーバ）

1. サーバ管理者は、キャッシュ再帰サーバを用いるための組織のセキュリティ方針があることを保証しなければならない。
2. 暗号鍵は適切に保護されなければならない（第一部を参照）。
3. DNS 検証キャッシュのトラストアンカーは、最新に維持されるべきである。

8.5.3 システム管理者のための推奨事項（クライアントシステム）

1. サーバ管理者は、クライアントシステムを使用するための組織のセキュリティ方針があることを保証しなければならない。

2. サーバ管理者は、利用者が適切に訓練されることを保証しなければならない。

8.6 エンドユーザのための推奨事項

エンドユーザとは、システムにアクセスするためにクライアントを使用する個人である。

1. エンドユーザは、彼らのクライアントシステムを彼らの組織およびシステム管理者の指示のとおり
に操作しなければならない。

9 暗号化ファイルシステム (EFS)

9.1 説明

データファイル及びディスクボリューム全体の暗号化は、ネットワーク通信されるデータの暗号化の問題よりも何らかの異なる鍵管理上の問題を提示する。ネットワーク通信セキュリティは、通信中の情報のプライバシーと完全性に焦点を当てているが、保管（例、ファイル）のセキュリティは永続的なデータのプライバシーと完全性およびこのデータのセキュアな共有に焦点を当てている。ファイル（訳注：論理的な保管単位）及びボリューム（訳注：物理的な保管単位）暗号化を取り巻く鍵管理ガイダンスは、1つのセクションにまとめるには十分類似している。

これまでのセクションとは異なり、プロトコルおよび／または標準がネットワークセキュリティコミュニティによって十分に研究されているところで、ファイル暗号化の商用ソリューションは、鍵を保管するための幅広くさまざまなセキュリティスキームと方法を利用している。このソリューションの範囲のため、本セクションは、洗練されていないが、ファイル暗号化のために使用される様々な方法を網羅するだろう。

ファイル暗号化システムの設計者が応えなければならない最も重要な質問は、以下のとおりである：

- ・ システムで鍵はどのように使用されるのか、鍵に対してどのような保護が提供されるのか？
- ・ システム上で鍵はどこに保管されるのか？
- ・ 数多くの利用者コミュニティのために、（非現実的な数の鍵が保管される必要なしに）その方法は規模を拡大できるのか？

9.1.1 必要とされる鍵の数

暗号化ファイルシステムについて、ファイルまたはファイルのグループを暗号化するために鍵が使用される。システムはそれぞれのファイルを別々の対称鍵で暗号化するか、または同じ対称鍵を使用して複数のファイルをまとめて暗号化するかのいずれかが可能である。最初ケースでは、共有する利用者に対してアクセスを提供することは、非常に容易である、例えば、単にその利用者に鍵を与えることにより。この方法においては、1つのファイルはシステムにおけるその他のファイルのいずれにもアクセスを提供しないで共有する利用者によってアクセスされることが可能であるだけである。この最初の方法の欠点は、多くのファイルが暗号化される場合、このモデルは直ちに扱いにくくなる可能性があることである、なぜなら、鍵はそれぞれのファイルについて要求され、共有する利用者に提供されなければならないからである。

2番目のケースでは、かなり少ない鍵が要求され、鍵配付処理を軽減する。しかし、共有する利用者が単一のファイルへのアクセスを要求するとき、その利用者にアクセスを与えることはより問題がある。共有する利用者へ単に鍵を送信することにより、その鍵を用いて暗号化された単一のファイルというよりも、その所有者⁴⁰のすべてのファイルへのアクセスが与えられてしまう。

しかし、上記の2番目のケースにおいて個別のファイルへのアクセスを許可するために使用可能なくつかの暗号管理アクションがある。この2番目のシステムでのアクセスを制限するための1つの選択肢は、所有者またはシステムがそのファイルを復号し共有する利用者へ送信するために（例、ネットワークセキュリティメカニズムとセッション鍵を用いて）新しい鍵を用いて再度暗号化することである。共通のファイルサーバを共有する両方の利用者のようなますます稀な場合、復号され再度暗号化されたファイルは共有利用者のファイル空間に置かれるだろう。これは、所有者と共有利用者の間、またはファイルシステム管理処理と共有利用者間の交換のための、重要な処理オーバーヘッドと鍵管理プロトコルを要求する。この処理は、そのファイルを保護する元の鍵と同じセキュリティ強度のこれらの新しい鍵の

⁴⁰ 所有者とは、鍵を共有するような個人または個人のグループまたは処理である可能性がある。

適切な保護を要求する。

別の選択肢は、共有する利用者に、暗号化されたファイル、および共有する利用者へ提供されたものを含めて所有者のファイルのすべてを復号するために使用可能な所有者の復号鍵が提供されることである。システムオーバーヘッドは軽減され、第三者管理者から鍵を保護することが可能かもしれないが、所有者は要求者が共通の鍵の下で保護されるすべてのファイルへのアクセスを許可するべきであることに合意するのはありそうもない。

前記の2つケースでより極端な例を提供したように、以下は、より一般的に使用されているアプローチである。この場合、システム上のそれぞれの利用者は、非対称鍵ペアを持っている。所有者のそれぞれのファイルは、異なるようにランダムに生成されたファイル暗号化対称鍵 (FEK) の下で暗号化される。FEK は、次に所有者の公開鍵を用いて暗号化され、暗号化ファイルと共に保管される。ファイルの所有者がこれを別の利用者と共有したいとき、所有者は暗号化された FEK を (所有者のプライベート鍵を用いて) 復号、次に共有する利用者の公開鍵を用いて FEK を暗号化する。暗号化されたファイルと再暗号化された FEK は、次に要求者へ提供されることが可能となる。このシステムはいくつかの長所がある。まず、所有者は1つの非対称鍵ペアのみを管理する必要がある。次に、それが利用者間でのより簡単なファイル共有を許容する。3つ目に、それが非常に有効である、なぜならファイルは共有するために再暗号化する必要がないからである。最後に、非対称鍵は所有者によって管理され、かつ FEK はファイルと共に暗号化された形で保管されるので、システムは任意の鍵を別々に管理する必要がない。

もちろん、所有者は異なるファイルまたは一連のファイルを暗号化するために異なる鍵を使用することができる。所有者が所与の鍵で暗号化するよう選択するより少ないファイルは、維持される必要のあるより多くの鍵や関連 (例、ファイルの同一性、ファイルグループ、個人の識別情報、またはアクセスグループの関連) をもたらす。

検討されなければならない暗号化ファイルシステム内の別の重要なコンセプトは、データ回復の実装方法である。システム内のデータ回復機能なしに利用者が鍵を喪失した場合、利用者のデータは永久に喪失される。このように、マスター管理者パスワードのようなデータ回復のいくつかの形式が含まれる⁴¹ことは、不可欠である。これは、ファイルを復号するための複数のパスワードを許容するようなファイル暗号化システムを要求する、パスワードのうち1つが利用者に提供され、2つめはマスター管理者パスワードの形式で管理者に提供される。別の可能性は、管理者が、利用者が自身のパスワードを喪失または忘却のときにのみを使用するために利用者のパスワードを保管する方法を持つことである。

システムの適用範囲が一組の個人から大きいネットワークまたはネットワーク間に拡張するとき、鍵の管理に関連する要素は扱いにくくなる可能性がある。大規模システムに関連する鍵管理の課題は以下を含む：

- ・ グローバルなデータ配置 (多数の所有者と大容量のデータ) に直面する背景での維持。
- ・ 管理および配付する非常に多くの鍵。
- ・ 大量の鍵が EFS 内の1つの場所に保管される場合、そのサイトが敵対者の魅力的な目標を提供する一大きなドメインの1つの信頼ポイント。
- ・ 失効された利用者 (組織を離れた個人、加入有効期限が到来した個人、またはさもなければ、ファイルへのアクセスがもはや許可されるべきでないような個人) のアカウント管理の困難さ。
- ・ 別の個人又は組織に対して保護されるデータの所有権の再割り当て。
- ・ 鍵を喪失したような事象におけるデータの回復 (例、組織を離れた個人によって暗号化され保管されていて見つけられないようなアーカイブされた暗号化されたデータの場合)。
- ・ 所有者の多くのファイルに対する鍵を提供され、次に鍵と所有者のデータをさらに別の利用者に提供するような共有する利用者。

⁴¹ データ回復がどのように達成されるかの詳細は、本文書の対象範囲を超える。

9.1.2 ファイル暗号化で使用される対称鍵へのアクセス

単一の対称鍵で保護されるファイルの数について決定がなされた後、鍵管理の質問が検討される。どのようにファイル暗号化システムは暗号鍵を生成するか？どのように鍵は保管され、保護されるか？このセクションは、これらの質問に答える共通の方法について、これらの強みと弱みを議論すると共に、明らかにする。技術の進歩に伴い、さらなる手法が開発され、以下の一覧は完全ではなく、必須と考えられるべきでもないようになる。

上記の重要な質問への共通の回答について検討する。まず、ファイル暗号化システムはどのように対称鍵を生成するか？1つの方法は、[\[SP 800-132\]](#)で記述されるとおり、パスワードから鍵を導出することである。この場合、パスワードのランダムさに依存するシステムのセキュリティ；通常、パスワードは鍵生成のために使用されるような十分なランダムさを含まない（即ち、それらは相対的に容易に推測可能である）。標準的な辞書攻撃はしばしば弱いパスワードを回復できるので、強いパスワードがこのタイプのシステムのセキュリティに不可欠である。システム内で鍵を生成する良好な乱数ビット生成を活用することが望ましい。承認された乱数ビット生成器は、[\[SP 800-90A\]](#)、[\[SP 800-90B\]](#)および[\[SP 800-90C\]](#)で見つけることでできる。

次に、鍵の保護方法の質問を検討する。コンピュータ上の鍵のセキュアな保管を開発するために **Trusted Computing Group (TCG)** によって多大な取り組みが行われている。この取り組みが成熟し続けるように、**Trusted Platform Module** チップは、その鍵キャッシュ管理を通して、EFS で使用される鍵を保護するための別のフォーマットを提供する。

次に、これらの鍵が保管される場合を検討する。鍵が保管される必要がある場合、それらはコンピュータ自身の上で、またはハードウェアトークン上で単に保管される可能性がある。あるいは、鍵は2つ（またはそれ以上）の要素に分割されて、例えば、1つの要素がハードウェアトークン上に保管され、その他の鍵要素がコンピュータ自身の上で保管される。分割鍵が採用される場合、鍵要素を結合するために使用される方法が重要となる：等しい長さの鍵要素上で **XOR** 操作を実行することは要素を単に結合するよりも良い。一般的なハードウェアトークンは、スマートカードを含む。ハードウェアトークンを用いる長所は、利用者がハードウェアトークンをコンピュータから離れたところに保管する場合、その鍵はトークンとコンピュータ間で分割され、敵対者は鍵を回復するために両方のハードウェアピースを回復する必要がある。追加のセキュリティとして、鍵分割を暗号化することによって適用されるかもしれない、おそらくパスワードを用いて。

上記質問への回答には多くの置換がある。これらの質問がどのように回答可能かについて、それぞれのシステムの賛否両論と共に、4つの例が検討されるだろう。その場合に望ましい具体的なタイプのシステムを通常指し示すような、ファイル暗号化システムが使用される具体的な環境を検討することが重要である。

検討される第一の例は、システム上のすべてのファイルを暗号化する単一の対称鍵を使用するようなファイル暗号化システムである。この単一の鍵は利用者のパスワードから[\[SP 800-132\]](#)の方法を用いて生成される。

2番目の例は、ハードディスク上に保管される、ファイルごとの暗号化鍵を活用するシステムである。鍵暗号化用鍵（各ファイルを暗号化する鍵を復号するためにも使用される）は、ハードドライブ（例、**Trusted Platform Module (TPM)** [\[TPM\]](#)）上にセキュアに保管される。

3番目の例は、ハードウェアトークン上に保管された1つの鍵コンポーネントとパスワードから導出されるその他の要素（例、鍵を導出するために[\[SP 800-132\]](#)の方法を用いて）と共に鍵を再作成するために **XOR** される2つの鍵要素へ分割されるような、ファイル毎の暗号化鍵を活用するシステムである。

4番目の例は、あらかじめ配布されたファイル所有者の非対称プライベート鍵によって暗号化された、

ファイル毎の暗号化鍵を使用する。先の三つ例はシステムの賛否両論をより明らかに示すことがとても困難である一方で、このシステムは現行のファイル暗号化パッケージと共通となる。

表 9-1：ファイル暗号化の例のまとめ

方法	賛	否
例 1： SP 800-132	<ul style="list-style-type: none"> - 最も安価なソリューション - 強いパスワードを活用することが合理的なセキュリティをもたらすことができる。 	<ul style="list-style-type: none"> - あまりセキュアでない、なぜならセキュリティはパスワードの強度にのみ依存するから。
例 2： 鍵暗号化用鍵	<ul style="list-style-type: none"> - コンピュータで直接ストレージをセキュアにする - 外部のソフトウェア攻撃と物理的脅威からセキュアにする 	<ul style="list-style-type: none"> - 相対的に新しい技術。 - 鍵は同じコンピュータ上にファイルとして保管される。
例 3： ハードウェアトークン	<ul style="list-style-type: none"> - 鍵を分割する。 - 復号には 2 つのハードウェアピースが必要。 - 正しく実装すれば高いセキュリティが得られる。 - ファイルまたはトークンが喪失した場合、ファイルはセキュアなままである。 	<ul style="list-style-type: none"> - より高価である。 - トークンが喪失すると、ファイルは復号できなくなる。
例 4： 利用者所有の 鍵暗号化用非対称鍵	<ul style="list-style-type: none"> - コンピュータに平文の鍵は一切保管されない。 - 効率のよいファイル共有。 - トークンが使用される場合、高度にセキュアである。 - 利用者のプライベート鍵の危殆化は、その利用者のファイルのみを危殆化する。 	<ul style="list-style-type: none"> - 利用者に自身の鍵ペアを管理することが必要。 - 利用者パスワードか利用者トークンのいずれかが必要。

9.2 セキュリティおよび適合性の問題

1. 任意の暗号化ファイルシステムは、連邦政府情報の保護に使用される場合、承認された暗号が採用されなければならない。
2. パスワードから導出される鍵は、オフラインの総当たり攻撃の困難さを最大化するために、強いパスワードを採用しなければならない[SP 800-118]。

9.3 調達担当官のための推奨事項

以下の推奨事項は EFS コンポーネントを取得するための調達決定を行うような任意の個人へ向けてのものである。

1. 利用者が自分のパスワードを忘却するまたは利用者利用可能でないような事象において、データが喪失しないように、データ回復機能（例、マスター管理者パスワード）を暗号化ファイルシステムが含むことを保証する。データ回復は、このタイプのシステムでは不可欠である。
2. パスワードから鍵を導出するような EFS システムは強いパスワードの使用を実施する機能を持っていることを保証する。
3. 暗号化ファイルシステムのセキュリティを増大させるために、システムは、ハードウェアトークンまたは TPM を鍵の保管のために使用するべきである。

9.4 システムインストーラのための推奨事項

システムインストーラとは、EFS コンポーネントをインストールし、コンポーネントの初期設定を行うような任意の個人である。

1. セキュリティのためにパスワードを活用するような EFS システムがインストールされる時、インストーラは、強いパスワードが EFS によって実施されることを要求しなければならない。これは、オフラインでの総当たり攻撃の困難さを最大化する。
2. システムインストーラは、システムのセキュリティを保証するために鍵のデータベースが暗号化によって保護されることを保証するべきである。さらに、鍵が EFS システムによって分割される場合、ハードウェアトークン上に保管される鍵要素は保護されなければならない。

9.5 システム管理者のための推奨事項

システム管理者とは、EFS を日々管理し、エンドユーザと対話するような任意の個人である。

1. システム管理者は、組織のセキュリティ方針が実施されることを保証しなければならない。
2. 鍵回復手順は、利用者が自身の認証情報（パスワード、トークンデータ等）を喪失する場合、利用者がそれらのデータを回復できることを保証するために実行されなければならない。これらの利用者を認証するためのデータ回復要員のための方法は、利用者の鍵またはファイルの回復の前に実行されるべきである。
3. EFS がシステム管理者によるデータ回復で使用するためのマスター管理者パスワードを含む場合、システム管理者は強いパスワードを活用しなければならない。
4. システム管理者は、最低限、パスワード、データ回復手順、及びシステム内で認証機能を活用するための彼らのシステムの利用者設定に焦点を当てたシステムのエンドユーザへ、訓練とセキュリティガイダンスを提供しなければならない。

9.6 エンドユーザのための推奨事項

エンドユーザとは、自らの情報をセキュアにそして共有するために EFS を使用する個人である。

1. 利用者選択されたパスワードが製品内部で使用される場合、エンドユーザはオフラインの総当たり攻撃の困難さを最大化するために強いパスワードを活用しなければならない。
2. エンドユーザは、暗号化ファイルシステム製品の使用に関してシステム管理者によって提供されるガイダンスに従わなければならない。
3. エンドユーザは、ハードウェアトークンを喪失、または、パスワードを忘却した場合、システム管理者に通知しなければならない。

10 セキュアシェル (SSH)

10.1 説明

SSH は、セキュアリモートログインとセキュアでないネットワークまたはインターネット越しのその他のセキュアネットワークサービスのためのクライアントとサーバ間のプロトコルである。インターネットエンジニアリングタスクフォース (IETF) は、SSH プロトコルを管理する[\[RFC 4251\]](#)⁴²。SSH プロトコルは、3つの主要な要素からなる：トランスポート層プロトコル [\[RFC 4253\]](#)、利用者認証プロトコル [\[RFC 4252\]](#) およびコネクションプロトコル [\[RFC 4254\]](#)。

10.1.1 トランスポート層プロトコル (SSH-TLP)

トランスポート層プロトコル (TLP) は、サーバ認証、機密性、および完全性を完全前方秘匿性 (訳注：PFS : Perfect Forward Secrecy)⁴³と共に提供する。SSH コネクションの確立は、使用されるアルゴリズムをネゴシエートし、サーバを認証する、TLP を用いて開始される。

プロトコルのアルゴリズムのネゴシエーションのステップは、鍵共有 (プロトコルでは鍵交換と呼ばれる)、サーバ認証、暗号とデータ完全性のためにクライアントとサーバによって使用されるアルゴリズムを決定するために使用される。クライアントからサーバへ、およびサーバからクライアントへの通信を保護するような暗号化とデータ完全性アルゴリズムは独立に選択される、すなわち、クライアントからサーバへの通信を保護する暗号化アルゴリズムは、サーバからクライアントへの通信を保護する案等価アルゴリズムと異なるかもしれない。

アルゴリズムのネゴシエーションが完了した後、TLP は選択された暗号アルゴリズムのために鍵と IV を供給するような鍵交換処理 [\[RFC 4253 セクション 8\]](#) においてクライアントに対してサーバを認証する。鍵交換処理は、サーバが公開鍵の所有者であることの保証を提供するが、サーバの同一性を検証するために追加のステップが要求されるかもしれない；サーバの公開鍵の検証に関する議論については、[セクション 10.2.1.2](#) を参照。

プロトコルは、サーバとクライアント間のそれぞれの方向の通信について MAC アルゴリズムを選択することによってデータ完全性保護を提供する。しかし、プロトコルは、このサービス (即ち、データ完全性保護) が無効化されることについても許容する。

10.1.2 利用者認証プロトコル (UAP)

利用者認証プロトコルは、サーバに対してクライアントを認証する。TLP が完了した後、サーバとクライアントは、ネゴシエーション処理からの選択された暗号アルゴリズムと鍵交換からの鍵を使用するような暗号化 SSH トンネル⁴⁴を用いて通信する ([セクション 10.1.1](#) の SSH TLP プロトコルの議論を参照) 暗号化 SSH トンネルを使用してサーバは、セキュアに任意のクライアント認証を行うことができる。UAP は、SSH コネクションプロトコルの 1 つの認証されたトンネルを提供する。

⁴² このセクションは、SSH2 または SSH-2 と呼ばれる SSH バージョン 2 について議論する。SSH1 は標準プロトコルには決してならない、また SSH2 プロトコルによって後に[\[RFC 4251\]](#)の SSH プロトコルとして置き換えられた。

⁴³ 完全前方秘匿性 (PFS) は、現在確立されたセッション鍵または長期間のプライベート鍵の危険化が、以前確立されセッション鍵の危険化を引き起こさないような鍵確立方法の暗号学的特性である。

⁴⁴ 暗号化トンネルは、コネクションを通してやり取りするデータトラフィックのすべてが暗号化されるときのエンドトゥエンドの通信コネクションである。

10.1.3 コネクションプロトコル (CP)

コネクションプロトコル [\[RFC 4254\]](#) は、SSH によって使用される暗号化トンネルをいくつかの論理チャネルへ多重化する。CP は、対話型ログインセッション、コマンドのリモート実行、転送 TCP/IP コネクション、および転送 X11 コネクション [\[X11\]](#) が、同時に確立された SSH トランスポート層コネクションと利用者認証コネクションを越えてどのように実行されることが可能かについて定義する。

10.2 セキュリティおよび適合性の問題

10.2.1 TLP 問題

10.2.1.1 アルゴリズムのネゴシエーション

TLPのこのステップでは、鍵交換（即ち、鍵共有）、公開鍵認証、データ暗号化とメッセージ認証に使用されるべきアルゴリズムが選択される。

1. SSHサーバとクライアントは、特定の暗号サービスのための両方の通信方向について、同じ NIST承認された暗号アルゴリズムを選択するべきである。例えば、同じ暗号アルゴリズムと MAC生成アルゴリズムは、両方の通信方向について機密性と完全性保護をそれぞれ提供するために使用されるべきである。

注釈：使用するために選択された暗号アルゴリズムは、サーバとクライアントの両方によって、またネゴシエーションで提供されるアルゴリズムリストのそれぞれにおけるこれらのアルゴリズムのためのクライアントの希望するレベルによって、提供されるアルゴリズムに依存する；クライアントの希望するレベルは、列挙されるアルゴリズムの順序によって示される。暗号アルゴリズムを選択するため定義された手順については、[\[RFC 4253\]](#)、[セクション 7.1](#) を参照。

2. スイートB 暗号アルゴリズムは、クライアントとサーバシステムの両方によってサポートされる場合、使用するよう推奨される。スイートB アルゴリズムは、[\[RFC 6239\]](#) で規定される。それらは：

- ・ 鍵共有（鍵交換）：ecdh-sha2-nistp256 および ecdh-sha2-nistp384 ([セクション 10.2.1.2](#) を参照)。
- ・ 公開鍵アルゴリズム（サーバとクライアント認証用）：x509v3ecdsa-sha2-nistp256 および x509v3-ecdsa-sha2-nistp384 ([セクション 10.2.1.3](#) を参照)。
公開鍵は X.509 バージョン 3 証明書で搬送される。
- ・ 暗号化と MAC：AEAD_AES_128_GCM および AEAD_AES_256_GCM ([セクション 10.2.1.4](#) と [10.2.1.5](#) を参照)。

10.2.1.2 鍵共有／鍵交換アルゴリズム

Diffie-Hellman 鍵共有／鍵交換アルゴリズムの2つのファミリーが、SSHで使用するために規定されている：有限体に基づくものと楕円曲線に基づくもの。

- ・ RFC 4253では、2つの有限体鍵交換方法について規定する："diffie-hellman-group1-sha1"と

"diffie-hellman-group14-sha1" [\[RFC 4253\]](#)。diffie-hellman-group1-sha1は、使用されてはならないような、112bits以下のセキュリティ強度を提供する1024-bit鍵を使用する。しかし、この場合、SHA-1の使用が許容可能であることに注意すること。

- RFC 4419では、2つの追加の有限体 Diffie-Hellman 鍵交換方法について規定する："diffie-hellman-group-sxchange-sha1"と"diffie-hellman-group-sxchange-sha256" [\[RFC 4419\]](#)。これらの2つの方法によってサポートされることが可能なmodulus長（即ち、鍵サイズ）は1024と8192の間であるが、少なくとも2048bitsのmodulus長が使用されなければならない。
- RFC 5656では、上記のオプションの代わりに使用可能な楕円曲線に基づく鍵共有オプションについて規定する[\[RFC 5656\]](#)。2つのオプションは、SSHで楕円曲線暗号をサポートしているときは、連邦政府向けに実装が必須である："ecdh-sha2-nistp256"と"ecdh-sha2-nistp384"。これらのオプションは、適切なNIST承認されたハッシュ関数と曲線と共に楕円曲線Diffie-Hellman の使用するように規定する："ecdh-sha2-nistp256"は、SHA-256とnistp256曲線の使用を規定し、"ecdh-sha2-nistp384"は、SHA384とnistp384曲線の使用を規定する [\[RFC 5656\]](#)。曲線についての情報は、[\[FISP 186-4\]](#) を参照。

10.2.1.3 公開鍵認証アルゴリズム

公開鍵認証アルゴリズムは、サーバ認証を行うために、[\[RFC 4253\]](#)、セクション8] で規定される鍵交換で使用されることが可能なデジタル署名アルゴリズムである。RFC 4253は、2つのデジタル署名アルゴリズムを規定する：RSAとDSA（RFCでは"ssh-dss"と呼ばれる）、サーバ認証のためのハッシュ関数としてSHA-1を使用する。[\[SP 800-131A\]](#)に従ってSHA-1はデジタル署名を生成することはもはや許容されないことに注意することが重要である。しかし、このプロトコルでは、署名関数（RSAかDSAのいずれか）の公開鍵サイズが少なくとも2048 bitsである限り、SHA-1がサーバ認証のために許容されている。

プロトコルは、サーバの公開鍵が検証なしで使用されることを許容する（即ち、公開鍵の有効性の保証を得ることなしに）。連邦政府で使用するためには、[\[SP 800-89\]](#)で要求されるとおり、デジタル署名検証が実行されることが可能である前に、クライアントは、公開鍵の有効性の保証を得なければならない。

SSHの追加の認証アルゴリズムは、[\[RFC 5656\]](#) で規定される。これらは、[\[FIPS 186-4\]](#) で規定される曲線を用いて、楕円曲線デジタル署名アルゴリズム（ECDSA）を使用する。表10-1は、連邦政府によって使用されるべき、ECDSAアルゴリズムと曲線の実装のための要求事項を識別する。

表 10-1：楕円曲線を用いる公開鍵認証方法

ECDSA	ハッシュアルゴリズム	IETF	連邦政府
ecdsa-sha2-nistp256	SHA-256	必須	必須
ecdsa-sha2-nistp384	SHA-384	必須	必須

[\[RFC 6187\]](#) は、デジタル署名アルゴリズムのために公開鍵を搬送するために X/509 バージョン 3 の使用を規定する。それは、SSH で使用する方法として、正式に SHA-256 を用いた 2048-bit RSA についても定義する。この組み合わせは 112 bits のセキュリティを提供し、許容される。認証のためにデジタル署名アルゴリズムと共に使用されるハッシュアルゴリズムが SSH TLP の鍵交換と鍵導出関数でのハッシュ関数 HASH と異なる可能性があることに注意することは重要である。

10.2.1.4 暗号アルゴリズム

SSH用にいくつかの暗号アルゴリズムが利用可能である；連邦政府のアプリケーションは、NIST承認されたアルゴリズムを使用しなければならない。現在規定されている承認された暗号アルゴリズムと利用モードは、3DES-CBC、AES-128-CBC、AES-192-CBCおよびAES-256-CBC [RFC 4253]、およびAEAD_AES_128_GCMおよびAEAD_AES_256_GCM [RFC 5647]。3DES-CBCがIETF実装では必須となっている。

プロトコルにおいて、暗号化されたトンネル（クライアントからサーバへ、またはサーバからクライアントへ）を保護するための暗号アルゴリズムとしてAEAD_AES_128_GCMまたはAEAD_AES_256_GCMのいずれかが選択されるとき、それはMACアルゴリズムとしても選択される。異なるアルゴリズムまたは一連のアルゴリズムがそれぞれの方向の通信で選択されるかもしれない。AEAD_AES_128_GCMまたはAEAD_AES_256_GCMが選択されるとき（暗号化とMAC生成のために）、アルゴリズムは1回のみ実行される（暗号化で1回と完全性保護で別の時というよりも）、なぜならこれらのモードは、同時に機密性と完全性保護の両方を提供するからである。

AEAD_AES_128_GCMまたはAEAD_AES_256_GCMが使用されるとき、SSHパケット長フィールドは暗号化されないが、追加の認証される（平文）データとして処理される。SSHでのこれらのアルゴリズムの使用についてのより詳細については、[RFC 5647]を参照。

10.2.1.5 メッセージ認証コード（MAC）アルゴリズム

いくつかのMACアルゴリズムが、SSHで利用可能である：連邦政府のアプリケーションは、NIST承認されたアルゴリズムを使用しなければならない。現在規定されているNIST承認されたSSH用MACアルゴリズムは、HMAC-SHA1（160 bitsのMACタグを持つ）。HMAC-SHA1-96（96 bitsのMACタグを持つ [RFC 4253]）、AEAD_AES_128_GCMおよびAEAD_AES_256_GCM [RFC 5647]。

セクション10.2.1.4で説明されるとおり、AEAD_AES_128_GCMまたはAEAD_AES_256_GCMは、暗号アルゴリズムとしても選択されるときにのみ、MACアルゴリズムとして選択されることが可能である。

10.2.1.6 ホスト公開鍵の検証

アルゴリズムネゴシエーションが完了した後、TLPは、鍵交換処理においてクライアントに対してサーバを認証する [RFC 4253、セクション 8]。この鍵交換において、2つの値を生成しつつ、Diffie-Hellman（DH）鍵交換が行われる：DH値⁴⁵ K および H と呼ばれる交換ハッシュ値。 H は、 K のハッシュ値および H の完全な仕様のためのその他のデータである [RFC 4253、セクション 8]。生成された H と K は、次に、選択された暗号アルゴリズムのIVと鍵を生成するための鍵導出関数への入力として使用される（この関数の仕様については [RFC 4253、セクション 7.2]を参照）。

サーバが実際に公開鍵（プロトコルではホスト公開鍵と呼ばれる）の所有者であることを証明するために、サーバは自分のプライベート鍵（プロトコルではホストプライベート鍵と呼ばれる）を用いて H （即ち、クライアントサーバの両方によって既知であるデータ）についてデジタル署名を生成する。クライアントがホスト公開鍵を用いてデジタル署名を検証できる場合、クライアントは、サーバが公開鍵の所有者であるという保証を得る。しかし、 H についての署名が検証される前に、クライアントによってホスト公開鍵が検証されることなしには、サーバの同一性は検証されない。

ホスト公開鍵の検証は、1) サーバ名とそのホスト公開鍵を信頼されるサーバ名とホスト公開鍵のデー

⁴⁵ [SP 800-56A]の鍵共有スキームでは、 K は共有秘密であるとみなされる。

データベースと比較すること、または 2) ホスト公開鍵証明書（即ち、サーバの公開鍵証明書）を用いること、または 3) SSH TLP⁴⁶の開始に際して DNSSEC を用いた out-of-band 検証方法を用いること、によって行われる。ホスト公開鍵とサーバ名の間に関連の検証は、SSH コネクションのセキュリティに必要不可欠なものである。コネクションが検証されない場合、コネクションは中間者攻撃の対象となる；この脆弱性の詳細は、[\[RFC 4251\]](#)で述べられている。ゆえに、ホスト公開鍵のサーバ名への対応は、TLP セッションごとに検証されなければならない、また検証が失敗した場合、TLP は失敗としなければならない。

- ・ サーバ認証の第一の方法が使用されるとき、それはクライアントのみが利用可能な方法であり、プロトコルは、一致したときのみ、継続しなければならない。ホスト公開鍵が認証されなるとき、プロトコルはコネクションが失敗することを明示的に要求しないが；この推奨事項はこのような場合にプロトコルの中止を要求する、ことに注意すること。
- ・ 2 番目の方法について、[\[RFC 4253\]](#) で規定される TLP は、クライアントが、a) 公開鍵とサーバ名の間に関連の検証なしに提示された公開鍵証明書を受け入れる、または b) ホスト公開鍵が検証される場合（公開鍵証明書にある公開鍵の検証法の詳細については[セクション 2](#)を参照）のみ証明書を検証し継続する、ことを許容する。連邦政府用途について、オプション b) はが使用されなければならない；即ち、この 2 番目の方法が使用されるとき、プロトコルはサーバの証明書が成功裏に検証されることなしに継続してはならない。
- ・ DNSSEC を用いた out-of-band 検証方法が使用されるとき（方法 3）、サーバの同一性と公開鍵は、ホスト公開鍵のフィンガープリント（ハッシュ値）がサーバの"SSHFP"リソースレコードにある公開鍵のフィンガープリントと一致することなしに受け入れられてはならない。"SSHFP"リソースレコードは、豪放なフィンガープリントとしてフィンガープリントが受け入れられる前に検証されなければならない。この方法についての詳細は、[\[RFC 4255\]](#)を参照。

その他のサーバ認証方法がプロトコルのために後で定義されるかもしれない。

共有秘密を生成するような[\[SP 800-56A\]](#)で期待される Diffie-Hellman プリミティブを含む[\[RFC 4253\]](#)、[セクション 8](#)で鍵交換が規定されることに注意することは重要である。[\[SP 800-56A\]](#)では、完全な鍵共有スキームは、鍵材料を導出するための共有秘密を使用するような具体的な鍵導出方法（例、鍵導出関数）を含む。SSH では、そのかわりに、[\[RFC 4253\]](#)、[セクション 7.2](#)で規定される鍵導出関数に対して、共有秘密が提供される。この鍵導出関数は、[\[SP 800-135\]](#)で承認される。ゆえに、[\[RFC 4253\]](#)、[セクション 7.2](#)における鍵導出関数と結合される、[\[RFC 4253\]](#)、[セクション 8](#)の Diffie-Hellman 鍵交換は、SSH TLP の承認された鍵共有方法である。

10.2.2 UAP 問題

サーバがクライアントを認証するために使用できるような多くの認証方法がある。サーバがサポートしなければならない必須の方法は、UAP [\[RFC 4252\]](#)、[セクション 7](#)の "publickey" と呼ばれる、公開鍵認証である。これは、SHA-1を用いたデジタル署名を除き、[セクション 10.2.1.3](#)で規定されるとおり、デジタル署名アルゴリズムの仕様を要求する。SHA-1デジタル署名は、クライアント認証のために使用されてはならない。どのようにクライアント認証がデジタル署名アルゴリズムの1つを用いて行われるかの具体的な詳細については、[\[RFC 4252\]](#)、[セクション 7](#)を参照。連邦政府のアプリケーションについて、クライアント認証のために使用されるデジタル署名アルゴリズムは、[\[SP 800-131A\]](#)で規定される要求事項を満たさなければならない。

"publickey"認証法では、有効なプライベート鍵の保持の証明は、クライアントの同一性の証明であるとみなされる。ゆえに、プライベート鍵を異なるクライアント（利用者）の間で共有することは禁止されている。しかし、現在の実践において、多くの組織は複数のクライアント（利用者）がプライベート鍵を共有することを許容している。この問題および同一性管理に関連するその他の問題に対処する

⁴⁶ DNSSEC の詳細については、[\[RFC 4255\]](#)で見つけられる。

ため、NISTは最近、Draft NISTIR 7966, *Security of Automated Access Management Using Secure Shell (SSH)* [[NISTIR 7966](#)]を発行した。

"publickey"方法について、2つのその他の認証方法がこのプロトコルのために定義されている。第一の方法はパスワードを使用することである。2番目の方法は、このSSHコネクションでサーバによって信頼されるようなホストシステムのプライベート鍵を用いることである；後者の方法は、"host-based authentication"（訳注：ホストベースの認証）と呼ばれる。

- ・ 連邦政府用について、パスワードを使用する認証は、TLPが暗号化トンネルを提供しない場合使用されてはならない、なぜならパスワードが平文で送信されてしまうからである。
- ・ "host-based authentication"方法 [[RFC 4252](#)、セクション9] について、クライアントは、サーバに対して認証を試行しているようなホストシステムの利用者である。この方法では、クライアントはホストの署名プライベート鍵を知っている。クライアントは、SSHコネクションにおいてサーバとの認証処理の間にこのホストを代行してデジタル署名を生成するためにこのプライベート鍵を使用する。サーバは、ホストシステムを認証するために、デジタル署名を検証する。認証が成功すると、クライアント（利用者）は、このホストと関連付けられた許可された利用者となり、次にこの利用者（クライアント）は、SSHサーバによって認証されたとみなされます。この"host-based authentication"方法は、クライアント認証のために使用されてはならない、なぜならこの方法は、直接クライアントの同一性の暗号学的保証をサーバへ提供しないからである-サーバはクライアントの正しい同一性を取得するためにホストシステムを信頼しなければならない。

また、Kerberos を用いたクライアントの認証（[セクション 6](#) 参照）が、[RFC 4462](#)においても記述されている。RFC 4462 が"GSS-API 鍵交換を用いた認証"と呼ばれる認証方法を規定する。この認証方法は、スタンドアロンの認証方法として使用されてはならない、なぜなら、この方法はクライアントの同一性を証明しないからである。

10.3 調達ガイダンス

以下の推奨事項は、SSHクライアントおよびサーバ実装を取得するための調達の決定をおこなうような任意の個人のためのものである。

1. クライアントとサーバの実装が少なくともこれらのNIST承認された暗号アルゴリズムの1つをサポートすることを保証する：プロトコルのための実装がすでに必須である、3DES-CBCに追加して、AES-128-CBC、AES-192-CBCおよびAES-256-CBC。これらの暗号アルゴリズムのすべてをサポートするような実装が選択されるべきである。
2. AEAD_AES_128_GCMまたはAEAD_AES_256_GCMが暗号アルゴリズムリストをネゴシエーション中に使用される場合、それが対応するMACアルゴリズムリストにもなければならぬ。
3. 実装は、上記の[セクション10.2.1.1](#)で記述されるとおり、スイートB暗号アルゴリズムをサポートするように選択されるべきである。
4. クライアントとサーバ実装が公開鍵証明書を用いた公開鍵認証をサポートすることを保証する。
5. クライアントとサーバ実装が、公開鍵証明書を検証が失敗するとき、プロトコルが中止されることを許容することを保証する。
6. クライアント実装が実装された暗号アルゴリズムのための優先設定を許容することを保証する。

7. サーバ実装が使用される暗号アルゴリズムの設定を許容することを保証する。
8. サーバ実装が、暗号がクライアントからサーバへのトラフィック方法で使用されないとき、パスワード認証方法を許容しないことが可能であることを保証する。
9. サーバ実装が"host-based authentication method"を許容しないことが可能であることを保証する。

10.4 システムインストーラのための推奨事項

システムインストーラとは、SSHサーバとクライアントアプリケーションをインストールし、システムの初期設定を行うような個人である。システムインストーラは以下を行わなければならない：

1. **承認された**暗号アルゴリズムのみを使用するようにサーバとクライアントをインストールおよび／または構成する。
2. 公開鍵証明書が両方の側：サーバとクライアント で検証されないとき、接続が失敗するよう設定する。
3. クライアントについて、組織のセキュリティガイドラインに従った暗号アルゴリズムリストのネゴシエーションにおける優先順位を設定する。例えば、組織がAES-256の仕様を望む場合、AES-256が利用可能な暗号選択肢のすべての中で最も高い優先度を持つように設定されなければならない。
4. サーバによって承認された暗号アルゴリズムのみの使用を設定する。
5. サーバがそのホストマシンのすべての利用者／クライアントとの通信を希望しない場合、そのサーバでの"host-based authentication"オプションを無効化しているとみなす。
6. AEAD_AES_128_GCMまたはAEAD_AES_256_GCMが暗号アルゴリズムリストのネゴシエーションにある時、自動化されていない場合、それが対となるMACアルゴリズムリストにもあるように設定する。
7. スイートB暗号アルゴリズムは、クライアント側で最も高い優先度となるよう設定されるべきである。これにより、サーバがスイートB 暗号アルゴリズムをサポートするとき、使用されるべきスイートB 暗号アルゴリズムが選択される結果となる。

インストーラは、SSHにおける暗号処理がデータトラフィックを保護するときに適切に機能できるように、すべての暗号要素と要求されるプラグインがインストールされることを確実にしなければならない。

10.5 システム管理者のための推奨事項

システム管理者とは、SSHサーバとクライアントアプリケーションが日々機能することに責任のある個人である。システム管理者は以下を行わなければならない：

1. TLPの鍵交換プロトコルでサーバの公開鍵証明書を検証し、検証失敗時にプロトコルを中止するようにクライアントを設定する。
2. 公開鍵アルゴリズムがUAPで使用される認証のために使用されるとき、クライアントの証明書が提供された場合、それを検証することによってクライアントを認証するようにサーバを設定する。

3. パスワードの選択、使用及び保護についての組織のセキュリティ方針に従うように、エンドユーザが適切に訓練されることを保証する。
4. 組織のセキュリティ方針が実行されることを保証する。
5. サーバの証明書およびクライアントの証明書に関連するプライベート鍵の、廃棄、漏えい、または許可されないアクセスからの保護が適切に設定されることを保証する。

10.6 エンドユーザのための推奨事項

エンドユーザとは、SSHサーバへセキュアに接続するためのSSHクライアントアプリケーションを用いる個人である。エンドユーザは、以下を行わなければならない：

1. その製品を使用するための組織のセキュリティ方針に従うよう周知され、訓練される。
2. 組織とシステム管理者によって指示されるとおりに彼らのシステムを操作する。

附属書 A：用語

以下に提供される用語は、本文書で使用されるために定義される。同じ用語がその他の文書では異なる定義であるかもしれない。

用語	定義
3-TDEA	[SP 800-67] で規定されるとおり 3 つの鍵の TDEA。
Access control アクセス制御	リソースへのアクセスを権限の与えられたエンティティだけに制限する
Access-control mechanism アクセス制御メカニズム	何らかのリソースへのアクセスを制限する方法
Approved 承認された	FIPS 承認されたおよび/または NIST 推奨のもの。 以下のいずれかのアルゴリズムまたは手法： 1) FIPS または NIST 推奨事項において規定された、または 2) FIPS または NIST 推奨事項で適用された、または 3) NIST 承認されたセキュリティ機能のリストで規定された
Archive (アーカイブ)	鍵管理アーカイブ を参照
Asymmetric-key algorithm 非対称鍵アルゴリズム	公開鍵暗号アルゴリズム を参照
Authentication 認証	情報の起源を確立する、またはエンティティの同一性を決定するような処理
Authentication code 認証コード	メッセージ認証コード を参照
Authorization 権限付与	エンティティに許可されるアクセス特権；所与のセキュリティ機能またはアクティビティを実行するための“公式の”許可を搬送する
Availability 可用性	情報への許可されたエンティティによるタイムリーで信頼できるアクセス
Backup バックアップ	必要に応じて、回復を促進するための情報の複製
CBC-MAC	ブロック暗号アルゴリズムの利用モードの 1 つ
Certificate (証明書)	公開鍵証明書 を参照
Certification authority 認証局	証明書の発行、および PKI ポリシーへの適合を確実にする責任を持つ公開鍵基盤 (PKI) におけるエンティティ
Checksum チェックサム	そのデータの誤りを検知することによってデータの完全性を保護するために使用される方法
Ciphertext (暗号文)	暗号化された形のデータ
Compromise	機微なデータの許可されない暴露、改ざん、置換または使用 (例. 鍵材料およびその他のセキュリティ関連の情報)。

危殆化

Confidentiality (機密性)	機微な情報が許可されないエンティティへ暴露されないような特性
Cryptographic key (key) 暗号鍵 (鍵)	鍵の知識を持つエンティティが再生成または操作を逆変換できるが、鍵の知識を持たないエンティティはできないような方法で操作を決定するような暗号アルゴリズムに関連して使用されるパラメータ。例として以下を含む： <ol style="list-style-type: none">1. 暗号文データへの平文のデータの変換2. 平文データへの暗号文データの変換3. データからデジタル署名の計算4. デジタル署名の検証5. データからの認証コードの計算6. データからの認証コードと受信された認証コードの検証 鍵材料を導出するために使用される共有秘密の計算
Cryptographic module 暗号モジュール	承認されたセキュリティ機能が実装されるハードウェア、ソフトウェア、および/またはファームウェアのセット
Cryptoperiod 暗号期間	特定の鍵の使用が承認されるか、与えられたシステムやアプリケーションにおいて鍵の有効性が残存するタイムスパン
Data integrity データ完全性	作成されるか送信するか格納された以降、承認されていない方法で、変更されていないデータ項目に付随した特性。 この推奨事項では、暗号アルゴリズムが“データ完全性を提供する”というステートメントは、無許可の変更を検知するためにアルゴリズムが使用されることを意味する
Decryption (復号)	暗号アルゴリズムと鍵を用いて、暗号文を平文に変換する処理
DES	FIPS 46 で規定されたデータ暗号化標準 (今は廃止されている)
Digital Signature デジタル署名	暗号学的変換の結果であり、支援する基盤および方針の下で適切に実現された時、次のサービスを提供する： <ol style="list-style-type: none">1. 作成者認証2. データ完全性3. 否認防止
Distribution (配付)	Key distribution (鍵配付) の項を参照
Encryption (暗号化)	暗号アルゴリズムと鍵を用いて、平文を暗号文に変更する処理。
Entity (エンティティ)	個人 (人)、グループ、デバイスまたはプロセス。
Hash algorithm ハッシュアルゴリズム	ハッシュ関数 を参照。
Hash-based message authentication code (HMAC) ハッシュベースメッセージ認証コード	承認された鍵付きハッシュ関数を用いるメッセージ認証コード。

Hash function ハッシュ関数	任意の長さのビット列を固定長ビット列に写像する機能。 承認されたハッシュ関数は次の特性を満たす： 1. (一方向) 任意のあらかじめ指定された出力に写像する入力を見つけることが計算上実行不可能であり、 2. (衝突困難性) 同一の出力に写像される 2 つ別個の入力の探索が計算上実行不可能であること。
Hash value (ハッシュ値) Identifier 識別子	情報に対してハッシュ関数を適用した結果。 人、デバイスまたは組織に関連付けられるビット列。 応用用途により、識別名やまたはより抽象的な列 (例. IP アドレスとタイムスタンプから成るビット列) となりうる。
Integrity 完全性	機微なデータが許可および検知されないようなやり方で改ざんまたは削除されるような特性。この推奨事項では、暗号アルゴリズムが“完全性を提供する”というステートメントは、そのアルゴリズムが許可されない改ざんや削除を検知するために使用されることを意味する。
Key (鍵)	暗号鍵 を参照。
Key agreement 鍵共有	どの当事者も鍵材料の値を事前に決定できないように、二者以上の関係者により与えられた情報の機能であるような鍵材料を合成するような鍵確立スキーム。
Key Bundle 鍵バンドル	TDEA の処理を典型として、一つの処理中に使用される一連の鍵。
Key component 鍵要素	平文の暗号鍵を形成したり、暗号機能を実行したりするため、承認されたセキュリティ機能の中で組み合わせて使用されるパラメータ。
Key distribution 鍵配付	鍵を所有または生成するエンティティからその鍵を使用すると意図される別のエンティティへの鍵およびその他の鍵材料の配送。
Key-encryption key 鍵暗号化用鍵	その他の鍵の暗号化または復号のために使用される暗号鍵。
Key establishment 鍵確立	異なる当事者間で共有されるような鍵材料を結果として生じる手順。
Key management 鍵管理	鍵の生成、保管、確立、入力及び出力、破壊を含め、鍵のライフサイクル全体で暗号鍵およびその他の関連セキュリティパラメータ (例. IV およびパスワード) の取扱いを包含するアクティビティ。
Key management archive 鍵管理アーカイブ	鍵材料のライフサイクルでの 1 つの機能；履歴として重要な鍵材料を含むリポジトリ。
Key pair 鍵ペア	公開鍵とその対応するプライベート鍵；鍵ペアは公開鍵アルゴリズムと共に使用される。
Key recovery 鍵回復	許可されたエンティティが鍵バックアップやアーカイブからの鍵材料の取り出しを許可するようなメカニズムとプロセス。

Key transport 鍵配送	1 つのエンティティ（送信者）が秘密の鍵材料としてある値を選択し、次にその値を別の当事者（受信者）へセキュアに配付するような鍵確立手順。
Key wrapping 鍵ラッピング	対称鍵を用いて機密性と完全性保護の両方を提供するような（関連する完全性情報と共に）鍵を暗号化する方法。
Key material 鍵材料	暗号学的な鍵の関係を確立し維持するために必要なデータ（例．鍵と IV）。
Message Authentication Code (MAC) メッセージ認証コード	偶然および意図的の両方のデータ改変を検知するために対称鍵を用いるデータ上の暗号学的チェックサム。
Non-repudiation 否認防止	主張される署名者のプライベート鍵の保持について特定のエンティティから作成されたとして、完全性と作成者が第三者によって検証可能であるような方法で、完全性とデータの作成者の保証を提供するために使用されるサービス。
Nonce ノンス	高々、無視できるほどの繰り返しとなるような、時刻と共に変化する値。例えば、ノンスとはランダムな値であり、ノンス、タイムスタンプ、シーケンス番号、またはこれらの組合せのそれぞれのインスタンスについて、新たに生成されるようなもの。
Owner 所有者	非対称鍵について、鍵ペアを生成したか、信頼される組織がそのエンティティの鍵ペアを生成したかのいずれかであるようなプライベート鍵を所有するエンティティ。 暗号化ファイルシステムにおいて、ファイル所有者は、ファイルについての制御を有し、そのファイルのアクセスをその他に許可する。1 つのファイルは、1 人または複数の所有者を持つかもしれない。所有者は、個人または個人のグループまたはプロセスである可能性がある。
Password パスワード	同一性を認証またはアクセス権限を検証するために使用されるような文字列（文字、数字及び記号）。
Payload ペイロード	通信における利用者情報および利用者オーバーヘッドをあらわすようなデータストリームの一部。
Plaintext（平文）	意味があり、理解可能な明瞭なデータ。
Private key プライベート鍵	暗号鍵：公開鍵暗号アルゴリズムと共に使用された、一意にエンティティに関連付けられ、公開されない。非対称（公開鍵）暗号系では、プライベート鍵は公開鍵と対応する。アルゴリズムに依存するが、プライベート鍵は、以下の例のように利用される： <ol style="list-style-type: none"> 1. 対応する公開鍵を算出する 2. 対応する公開鍵によって検証されるデジタル署名を作成する 3. 対応する公開鍵によって暗号化された鍵を復号する 4. 鍵共有処理において共有秘密を計算する。
Protocol プロトコル	エンティティ間で交換される情報についてのメッセージ順序およびデータ構造 1 つまたは複数のエンティティによって使用される規則の特別なセット。

<p>Public key 公開鍵</p>	<p>公開鍵暗号アルゴリズムで使用される暗号鍵で、エンティティと一意に関連付けられ、公開されるもの。非対称鍵（公開鍵）暗号系では、公開鍵はプライベート鍵に対応している。</p> <p>公開鍵は不特定多数に知らされ得るもので、アルゴリズムに依存し、例えば、以下のような用途で使用できる：</p> <ol style="list-style-type: none"> 1. 対応するプライベート鍵により署名されたデジタル署名を検証する 2. 対応するプライベート鍵を用いて復号可能な鍵を暗号化する 3. 鍵共有トランザクションにおいて共有秘密を計算する。
<p>Public-key (asymmetric) cryptographic algorithm 公開鍵（非対称）暗号アルゴリズム</p>	<p>2つの関連する鍵（公開鍵とプライベート鍵）を使用する暗号アルゴリズム。（公開鍵とプライベート鍵の）2つの鍵には、公開鍵からプライベート鍵を決定することは計算量的に困難であるという特性がある。</p>
<p>Public-key certificate 公開鍵証明書</p>	<p>エンティティを一意に識別する一連のデータで、エンティティの公開鍵や他の情報を含み、信頼される組織によりデジタル署名され、その結果公開鍵をエンティティに結合する。証明書の追加情報として鍵用途と鍵暗号周期を規定できる。</p>
<p>Public key Infrastructure (PKI) 公開鍵基盤</p>	<p>公開鍵証明書を発行し、維持し、かつ失効するために確立されるフレームワーク。</p>
<p>Reconstitute 再構築</p>	<p>単にバックアップからシステムを再スタートするのではなく、以前保存されたセキュリティ情報および/または鍵材料を用いてサービスシステムを再構築すること。</p>
<p>Rekey（鍵再作成）</p>	<p>新しい鍵で別の鍵を置換する；新しい鍵の“値”は、古い鍵の“値”と完全に独立である。</p>
<p>Relying Party 依拠当事者</p>	<p>利用者のアイデンティティ；公開鍵、対応するアルゴリズムと関連パラメータの有効性；及び対応するプライベート鍵の利用者保持、を確認するために、証明書とその証明書を発行したCA（認証局）に依拠する個人または組織。</p>
<p>Secret key 秘密鍵</p>	<p>一つ以上のエンティティに関連付けられ、かつ非公開であるような、秘密鍵（対称鍵）暗号アルゴリズムで使用される暗号鍵。この文脈での“秘密”という用語の使用は、格付けレベルを意味しないが、むしろ暴露から鍵を保護する必要性を示している。</p>
<p>Security association セキュリティアソシエーション</p>	<p>セキュリティ情報が共有され、2つのエンティティ間のセキュアな通信をサポートするような、2つのネットワークエンティティ間の関係。</p>
<p>Security services セキュリティサービス</p>	<p>機密性、データ完全性、認証または情報の否認防止を提供するために使用されるメカニズム。</p>
<p>Self-signed certificate 自己署名証明書</p>	<p>証明書内に含まれる公開鍵によって検証されるデジタル署名を持つ公開鍵証明書。自己署名された証明書上の署名は、情報のデータ完全性を提供するが、情報の真正性を保証しない。自己署名証明書の信頼は、それらを配付する安全な手順に基づくものである。</p>

<p>Shall しなければならない</p>	<p>この用語は連邦情報処理標準 (FIPS) の要求、または本推奨事項への適合を主張する際に満たされなければならない要求を示すために使用される。</p> <p>“しなければならない (Shall) ” の否定は、“してはならない (Shall not)” となることに注意する。</p>
<p>Shared secret 共有秘密</p>	<p>鍵共有スキームで計算される秘密の値で有り、鍵導出関数/方法への入力として使用される</p>
<p>Should すべきである</p>	<p>この語は、非常に重要な推奨事項を表す。推奨事項を無視することは好ましくない結果を招来しかねない。”～すべきである”の否定は“～すべきでない (should not)” となる</p>
<p>Signature verification 署名検証</p>	<p>デジタル署名アルゴリズムと公開鍵を用いて、データに対するデジタル署名を検証する</p>
<p>Symmetric key 対称鍵</p>	<p>対称鍵 (暗号) アルゴリズムで使用される単一の鍵</p>
<p>Symmetric-key algorithm 対称鍵アルゴリズム</p>	<p>(例. 暗号化と復号のような) 単一の処理でかつその処理が完了するまで同じ共通鍵を使用する暗号アルゴリズム</p>
<p>Threat 脅威</p>	<p>データの許可されないアクセス、破壊、暴露、改ざんまたはサービスの拒否を通じて、システムに影響を受ける可能性のある任意の環境または事象。</p>
<p>Triple DES/Triple DEA (TDEA) トリプル DES/トリプル DEA</p>	<p>[SP 800-67] で規定された、トリプルデータ暗号化アルゴリズム</p>
<p>Unauthorized disclosure 許可されない暴露</p>	<p>当該情報へのアクセスを許可されていないエンティティへの情報の開示に関する事象</p>
<p>User's name (in a certificate) (証明書内の) 利用者名称</p>	<p>証明書内の公開鍵；証明書のサブジェクトに関連付けられるプライベート鍵を使用することを許可された当事者の名前。</p>
<p>X.509 public-key certificate X.509 公開鍵証明書</p>	<p>ITU-T X.509 標準において定義されたフォーマットでエンコードされて、証明書を発行した認証局のデジタル署名により偽造不可能とされる、他の情報と共に利用者 (またはデバイス) の公開鍵および利用者 (またはデバイス) の名称を含むデジタルの証明書</p>

附属書 B : 略語

AES	Advanced Encryption Standard
AH	Authentication Header
AS	Authentication Server
CA	Certificate Authority
CBC	Cipher Block Chaining
CBC-MAC	Cipher Block Chaining Message Authentication Code
CMVP	Cryptographic Module Validation Program
COTS	Commercial Off-the-Shelf
CRL	Certificate Revocation List
CTR	Counter Mode
DES	Data Encryption Standard
DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DSA	Digital Signature Algorithm
EC	Elliptic Curve
ECDH	Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
EFS	Encrypted File System
ESP	Encapsulating Security Protocol
FEK	File Encryption Key
GCM	Galois Counter Mode
GMAC	Galois Message Authentication Code
HMAC	Hash-based Message Authentication Code
HTTP	Hypertext Transfer Protocol
ICV	Integrity Check Value
IETF	Internet Engineering Task Force
IKE	Internet Key Exchange
IP	Internet Protocol
IV	Initialization Vector
KDC	Key Distribution Center
KMF	Key Management Facility
KMM	Key Management Message
KSK	Key Signing Key
KWK	Key Wrapping Key
LDAP	Lightweight Directory Access Protocol
MAC	Message Authentication Code

MD5	Message-Digest Algorithm 5
NIST	National Institute of Standards and Technology
NSEC	Next Secure
OCSP	Online Certificate Status Protocol
OMB	Office of Management and Budget
OTAR	Over The Air Rekeying
PIN	Personal Identification Number
PKCS	Public Key Cryptography Standard
PKI	Public Key Infrastructure
PRF	Pseudorandom Function
PVM	Path Validation Module
RA	Registration Authority
RFC	Request for Comment
RR	Resource Record
RSA	Rivest, Shamir, and Adleman.
S/MIME	Secure/Multipart Internet Mail Extensions
SA	Security Association
SEP	Secure Entry Point
SHA	Secure Hash Algorithm
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
TCG	Trusted Computing Group
TDEA	Triple Data Encryption Algorithm
TEK	Traffic Encryption Key
TGS	Ticket Granting Service
TLS	Transport Layer Security
TPM	Trusted Platform Module
TS	Target Server
TSIG	Transaction Signature
URL	Uniform Resource Locator
VPN	Virtual Private Network
XOR	Exclusive-Or operation
ZSK	Zone Signing Key

附属書 C：初心者エンドユーザへの言葉

暗号鍵は、使用されるアルゴリズム、実行時に使用される操作および使用可能な回数によってしばしば分類される。鍵は、非対称で非対称アルゴリズムと共に使用されるか、対称で対称アルゴリズムと共に使用されるか、のいずれかである。非対称鍵は鍵ペアとして生成される：秘密に保たなければならないようなプライベート鍵、および関連する秘密でないかもしれない公開鍵。一般にプライベート鍵は、例えばデジタル署名のような、1つの操作を行い、公開鍵は補完的な操作、この場合は署名検証のような操作を行う。対称鍵の場合は、エンドユーザは秘密の共有の値のように鍵を取り扱う必要があり、同じ値を、暗号化と復号のような、相補的な暗号機能のために使用する。

いくつかの非対称鍵は静的で、長期間の使用を意図しているが、その他は一時的であり、1つのメッセージまたはセッションで使用した後、期限切れとなる。静的な非対称鍵ペアの公開鍵は、しばしば公開鍵証明書で提供されるが、一時的な公開鍵はこのようには提供されない。静的対一時的の概念は対称鍵に適用されるが、短期の対称鍵は、一時的鍵ではなく、しばしばセッション鍵と呼ばれる。アプリケーションまたはプロトコルはこのような鍵のいくつかの組合せによってサポートされるかもしれない。

PKI は多くの現在の鍵管理プロセスの基盤であり、その他のセキュリティプロトコル及びアプリケーションと同様に、本推奨事項で記述されるプロトコルまたはアプリケーションの多くで使用されている。PKI の役割の何らかの理解と鍵管理における公開鍵証明書は、セキュリティプロトコルまたはアプリケーションを適切に設定するために非常に役立つ。長期間、または“静的”公開鍵は一般に公開鍵証明書と呼ばれる電子的な文書において鍵の“所有者”という名称と結合される。証明書は自分で発行され署名されることもできるが（即ち、鍵ペアを生成した当事者が自身の名前とプライベート鍵に対応する公開鍵に署名することができる）、ほとんどの証明書は、認証局と呼ばれる信頼されるエンティティのプライベート鍵でデジタル的に署名される。

エンドユーザは一般に1つか複数の証明書を持っているかもしれないし、異なるアプリケーションのために異なる複数の証明書を持っているかもしれない、例、電子メール用、およびウェブサイトへの認証用。連邦政府個人 ID 検証 (PIV) カードは連邦政府の職員及び嘱託職員に発行され、ほとんどの連邦政府利用者は、かれらの機関によって発行される1つまたは複数の個人の証明書を含むような個人のスマートカードを持っている。その他の具体的なアプリケー所が“ソフトな”証明書を要求するかもしれない、通常利用者コンピュータ内に保持され、商用 CA によって発行されるかもしれない。例えば、インターネットエクスプローラ、Firefox、または Safari のような、鍵ペアを生成するメカニズムを実装し、CA に公開鍵を送信し、その鍵の公開鍵証明書をブラウザに返送するようなブラウザ製品。証明書は利用者のコンピュータの利用者証明書ストアに保管され、ルート証明書ストアに似たやり方で管理することができる（[セクション 2](#)を参照）。鍵を生成し、証明書を要求、ダウンロード、インストールするための処理とインタフェースは、個別の利用者クライアント製品特有であり、時には CA 自体に特有かも知れない；しかし CA のウェブサイトは一般のブラウザ製品のための鍵生成と証明書発行処理を通して利用者を案内するようなページをしばしば持っている。利用者は、電子メールや公開鍵基盤、スマートカードまたはその他のメモリートークンを経由して証明書を共有することができる。

同様に、セキュアなウェブサーバは、一定の具体的な特徴を持つ、TLS サーバ証明書を持っている：連邦政府の利用者は、SSL よりむしろ、TLS を使用することが要求されるが、証明書は同一であることに注意すること。これらの証明書のサブジェクト名は、サーバのドメイン名が証明書のサブジェクト名フィールドに含まれるように具体的な規則に従っている。商用 CA は TLS 証明書を販売し、また非政府利用者の母集団に到達することは重要であるように、マイクロソフトウィンドウズ、マッキントッシュ OS X、およびさまざまな Mozilla ブラウザの証明書ストアに“買ったその日から使える”ように幅広く配付されたルート証明書を持つような認証局から TLS 証明書を取得することが望ましいかもしれない。これはほとんどの利用者がサーバ証明書の検証を可能にする。しかし、本書および[\[SP 800-57 Part 1\]](#)で述べた要求事項を証明書が満たすことを保証するために連邦政府ウェブサーバ上で使用するための TLS 証明書を販売する CA から利用可能にするように、証明書ポリシーを見直し、選択することは、重要である。

附属書 D : 参考文献

- [COMMON] *X.509 Certificate Policy for the U.S. Federal PKI Common Policy Framework, Version 1.23*, Federal Public Key Infrastructure Policy Authority, May 5, 2014, 105 pp
http://www.idmanagement.gov/sites/default/files/documents/FCPCA%20CP%20v1%2023_0.pdf [accessed 12/23/14].
- [COMMON PROF] *X.509 Certificate and Certificate Revocation List (CRL) Extensions Profile for the Shared Secret Providers (SSP) Program*, Federal PKI Policy Authority, Shared Service Provider Working Group, January 7, 2008, 52 pp.
<http://www.idmanagement.gov/sites/default/files/documents/CertCRLprofileForCP.pdf> [accessed 12/23/14].
- [DENN] D. Denning and G. M. Sacco, “Timestamps in Key Distribution Protocols,” *Communications of the ACM*, vol. 24, no. 8, pp. 533-536, August 1981.
<http://dx.doi.org/10.1145/358722.358740>.
- [FIPS 140-2] U.S. Department of Commerce, *Security Requirements for Cryptographic Modules*, Federal Information Processing Standards (FIPS) Publication 140-2, National Institute of Standards and Technology, May 25, 2001 (including change notices as of December 3, 2002) , 69 pp.
<http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf> [accessed 12/23/14].
- [FIPS 180-4] U.S. Department of Commerce, *Secure Hash Standard (SHS)* , Federal Information Processing Standards (FIPS) Publication 180-4, National Institute of Standards and Technology, March 2012, 37 pp. <http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf> [accessed 12/23/14].
- [FIPS 186-4] U.S. Department of Commerce, *Digital Signature Standard (DSS)* , Federal Information Processing Standards (FIPS) Publication 186-4, National Institute of Standards and Technology, July 2013, 130 pp.
<http://dx.doi.org/10.6028/NIST.FIPS.186-4>.
- [FIPS 197] U.S. Department of Commerce, *Advanced Encryption Standard (AES)* , Federal Information Processing Standards (FIPS) Publication 197, National Institute of Standards and Technology, November 2001, 51 pp.
<http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf> [accessed 12/23/14].
- [FIPS 201] U.S. Department of Commerce, *Personal Identity Verification (PIV) of Federal Employees and Contractors*, Federal Information Processing Standards (FIPS) Publication 201-2, National Institute of Standards and Technology, August 2013, 87 pp. <http://dx.doi.org/10.6028/NIST.FIPS.201-2>.
- [FPKI PROF] *Federal Public Key Infrastructure (PKI) X.509 Certificate and CRL Extensions Profile*, Booz-Allen & Hamilton, Inc., and National Institute of Standards and

Technology, October 12, 2005, 33 pp.

http://www.idmanagement.gov/sites/default/files/documents/fpki_certificate_profile_0.pdf [accessed 12/23/14].

- [NEED] R. M. Needham and M. D. Schroeder, “Using Encryption for Authentication in Large Networks of Computers,” *Communications of the ACM*, vol. 21, no. 12, pp. 993-999, December 1978. <http://dx.doi.org/10.1145/359657.359659>.
- [NEUM] B. C. Neuman and T. Y. Ts'o, “Kerberos: An Authentication Service for Computer Networks,” *IEEE Communications Magazine*, vol. 32, no. 9, pp. 33-38, September 1994. <http://dx.doi.org/10.1109/35.312841>.
- [NISTIR 7966] T. Ylonen, K. Scarfone and M. Souppaya, *Security of Automated Access Management Using Secure Shell (SSH)*, National Institute of Standards and Technology, (DRAFT) NISTIR 7966, August 2014, 43 pp. <http://csrc.nist.gov/publications/PubsNISTIRs.html> [accessed 12/23/14].
- [OMB 04-04] Office of Management and Budget (OMB), *E-Authentication Guidance for Federal Agencies*, OMB Memorandum M-04-04, December 16, 2003, 17 pp. <http://www.whitehouse.gov/sites/default/files/omb/assets/omb/memoranda/fy04/m04-04.pdf> [accessed 12/23/14].
- [OTAR] *Project 25— Digital Radio Over-the-Air-Rekeying (OTAR) Protocol*, TIA/EIA 102.AACA, April 2001.
- [OTAR1] *Project 25—Digital Radio Over-the-Air-Rekeying (OTAR) Protocol—Addendum 1— Key Management Security Requirements for Type 3 Block Encryption Algorithms*, TIA/EIA 102.AACA-1, November 2002.
- [RFC 1034] P. Mockapetris, *Domain Names – Concepts and Facilities*, Internet Engineering Task Force (IETF) RFC 1034, November 1987. <http://www.ietf.org/rfc/rfc1034.txt> [accessed 12/23/14].
- [RFC 1035] P. Mockapetris, *Domain Names – Implementation and Specification*, Internet Engineering Task Force (IETF) RFC 1035, November 1987. <http://www.ietf.org/rfc/rfc1035.txt> [accessed 12/23/14].
- [RFC 1510] J. Kohl and C. Neuman, *The Kerberos Network Authentication Service (V5)*, Internet Engineering Task Force (IETF) RFC 1510, September 1993. <http://www.ietf.org/rfc/rfc1510.txt> [accessed 12/23/14].
- [RFC 2045] N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, Internet Engineering Task Force (IETF) RFC 2045, November 1996. <http://www.ietf.org/rfc/rfc2045.txt> [accessed 12/23/14].
- [RFC 2046] N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part 2: Media Types*, Internet Engineering Task Force (IETF) RFC 2046, November 1996. <http://www.ietf.org/rfc/rfc2046.txt> [accessed 12/23/14].
- [RFC 2047] K. Moore, *MIME (Multipurpose Internet Mail Extensions) Part Three: Message*

- Header Extensions for Non-ASCII Text*, Internet Engineering Task Force (IETF) RFC 2047, November 1996. <http://www.ietf.org/rfc/rfc2047.txt> [accessed 12/23/14].
- [RFC 2049] N. Freed and N. Borenstein, *Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples*, Internet Engineering Task Force (IETF) RFC 2049, November 1996. <http://www.ietf.org/rfc/rfc2049.txt> [accessed 12/23/14].
- [RFC 2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, Internet Engineering Task Force (IETF) RFC 2119, March 1997. <http://www.ietf.org/rfc/rfc2119.txt> [accessed 12/23/14].
- [RFC 2246] T. Dierks and C. Allen, *The TLS Protocol Version 1.0*, Internet Engineering Task Force (IETF) RFC 2246, January 1999 (obsoleted by RFC 4346) . <http://www.ietf.org/rfc/rfc2246.txt> [accessed 12/23/14].
- [RFC 2401] S. Kent and R. Atkinson, *Security Architecture for the Internet Protocol*, Internet Engineering Task Force (IETF) RFC 2401, November 1998 (obsoleted by RFC 4301) . <http://www.ietf.org/rfc/rfc2401.txt> [accessed 12/23/14].
- [RFC 2402] S. Kent and R. Atkinson, *IP Authentication Header*, Internet Engineering Task Force (IETF) RFC 2402, November 1998 (obsoleted by RFC 4302 and RFC 4835) . <http://www.ietf.org/rfc/rfc2402.txt> [accessed 12/23/14].
- [RFC 2404] C. Madson and R. Glenn, *The Use of HMAC-SHA-1-96 within ESP and AH*, Internet Engineering Task Force (IETF) RFC 2404, November 1998. <http://www.ietf.org/rfc/rfc2404.txt> [accessed 12/23/14].
- [RFC 2405] C. Madson and N. Doraswamy, *The ESP DES-CBC Cipher Algorithm with Explicit IV*, Internet Engineering Task Force (IETF) RFC 2405, November 1998. <http://www.ietf.org/rfc/rfc2405.txt> [accessed 12/23/14].
- [RFC 2406] S. Kent and R. Atkinson, *IP Encapsulating Security Payload (ESP)* , Internet Engineering Task Force (IETF) RFC 2406, November 1998 (obsoleted by RFC 4303 and RFC 4835) . <http://www.ietf.org/rfc/rfc2406.txt> [accessed 12/23/14].
- [RFC 2407] D. Piper, *The Internet IP Security Domain of Interpretation for ISAKMP*, Internet Engineering Task Force (IETF) RFC 2407, November 1998 (obsoleted by RFC 5996) . <http://www.ietf.org/rfc/rfc2407.txt> [accessed 12/23/14].
- [RFC 2408] D. Maughan, M. Schertler, M. Schneider, and J. Turner, *Internet Security Association and Key Management Protocol (ISAKMP)* , Internet Engineering Task Force (IETF) RFC 2408, November 1998 (obsoleted by RFC 5996) . <http://www.ietf.org/rfc/rfc2408.txt> [accessed 12/23/14].
- [RFC 2409] D. Harkins and D. Carrel, *The Internet Key Exchange (IKE)* , Internet Engineering Task Force (IETF) RFC 2409, November 1998 (obsoleted by RFC 5996) . <http://www.ietf.org/rfc/rfc2409.txt> [accessed 12/23/14].
- [RFC 2410] R. Glenn and S. Kent, *The NULL Encryption Algorithm and its Use with IPsec*,

- Internet Engineering Task Force (IETF) RFC 2410, November 1998.
<http://www.ietf.org/rfc/rfc2410.txt> [accessed 12/23/14].
- [RFC 2451] R. Pereira and R. Adams, *The ESP CBC-Mode Cipher Algorithms*, Internet Engineering Task Force (IETF) RFC 2451, November 1998.
<http://www.ietf.org/rfc/rfc2451.txt> [accessed 12/23/14].
- [RFC 2631] E. Rescorla, *Diffie-Hellman Key Agreement Method*, Internet Engineering Task Force (IETF) RFC 2631, June 1999. <http://www.ietf.org/rfc/rfc2631.txt> [accessed 12/23/14].
- [RFC 2634] P. Hoffman (ed.) , *Enhanced Security Services for S/MIME*, Internet Engineering Task Force (IETF) RFC 2634, June 1999. <http://www.ietf.org/rfc/rfc2634.txt> [accessed 12/23/14].
- [RFC 3394] J. Schaad and R. Housley, *Advanced Encryption Standard (AES) Key Wrap Algorithm*, Internet Engineering Task Force (IETF) RFC 3394, September 2002.
<http://www.ietf.org/rfc/rfc3394.txt> [accessed 12/23/14].
- [RFC 3447] J. Jonsson and B. Kaliski, *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*, Internet Engineering Task Force (IETF) RFC 3447, February 2003. <http://www.ietf.org/rfc/rfc3447.txt> [accessed 12/23/14].
- [RFC 3566] S. Frankel and H. Herbert, *The AES-XCBC-MAC-96 Algorithm and its Use with IPsec*, Internet Engineering Task Force (IETF) RFC 3566, September 2003.
<http://www.ietf.org/rfc/rfc3566.txt> [accessed 12/23/14].
- [RFC 3602] S. Frankel, R. Glenn and S. Kelly, *The AES-CBC Cipher Algorithm and its Use with IPsec*, Internet Engineering Task Force (IETF) RFC 3602, September 2003.
<http://www.ietf.org/rfc/rfc3602.txt> [accessed 12/23/14].
- [RFC 3645] S. Kwan, P. Garg, J. Gilroy, L. Esibov, J. Westhead and R. Hall, *Generic Security Service Algorithm for Secret Key Transaction Authentication for DNS (GSS-TSIG)* , Internet Engineering Task Force (IETF) RFC 3645, October 2003.
<http://www.ietf.org/rfc/rfc3645.txt> [accessed 12/23/14].
- [RFC 3686] R. Housley, *Using Advanced Encryption Standard (AES) Counter Mode with IPsec Encapsulating Security Payload (ESP)* , Internet Engineering Task Force (IETF) RFC 3686, January 2004. <http://www.ietf.org/rfc/rfc3686.txt> [accessed 12/23/14].
- [RFC 3962] K. Raeburn, *Advanced Encryption Standard (AES) Encryption for Kerberos 5*, Internet Engineering Task Force (IETF) RFC 3962, February 2005.
<http://www.ietf.org/rfc/rfc3962.txt> [accessed 12/23/14].
- [RFC 4033] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, *DNS Security Introduction and Requirements*, Internet Engineering Task Force (IETF) RFC 4033, March 2005.
<http://www.ietf.org/rfc/rfc4033.txt> [accessed 12/23/14].
- [RFC 4034] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, *Resource Records for the DNS Security Extensions*, Internet Engineering Task Force (IETF) RFC 4034,

- March 2005. <http://www.ietf.org/rfc/rfc4034.txt> [accessed 12/23/14].
- [RFC 4035] R. Arends, R. Austein, M. Larson, D. Massey and S. Rose, *Protocol Modifications for the DNS Security Extensions*, Internet Engineering Task Force (IETF) RFC 4035, March 2005. <http://www.ietf.org/rfc/rfc4035.txt> [accessed 12/23/14].
- [RFC 4106] J. Viega and D. McGrew, *The Use of Galois/Counter Mode (GCM) in IPsec Encapsulating Security Payload (ESP)*, Internet Engineering Task Force (IETF) RFC 4106, June 2005. <http://www.ietf.org/rfc/rfc4106.txt> [accessed 12/23/14].
- [RFC 4109] P. Hoffman, *Algorithms for Internet Key Exchange version 1 (IKEv1)*, Internet Engineering Task Force (IETF) RFC 4109, May 2005. <http://www.ietf.org/rfc/rfc4109.txt> [accessed 12/23/14].
- [RFC 4120] C. Neuman, T. Yu, S. Hartman and K. Raeburn, *The Kerberos Network Authentication Service (V5)*, Internet Engineering Task Force (IETF) RFC 4120, July 2005. <http://www.ietf.org/rfc/rfc4120.txt> [accessed 12/23/14].
- [RFC 4210] C. Adams, S. Farrell, T. Kause and T. Mononen, *Internet X.509 Public Key Infrastructure Certificate Management Protocol (CMP)*, Internet Engineering Task Force (IETF) RFC 4210, September 2005. <http://www.ietf.org/rfc/rfc4210.txt> [accessed 12/23/14].
- [RFC 4251] T. Ylonen and C. Lonvick (ed.), *The Secure Shell (SSH) Protocol Architecture*, Internet Engineering Task Force (IETF) RFC 4251, January 2006. <http://www.ietf.org/rfc/rfc4251.txt> [accessed 12/23/14].
- [RFC 4252] T. Ylonen and C. Lonvick (ed.), *The Secure Shell (SSH) Authentication Protocol*, Internet Engineering Task Force (IETF) RFC 4252, January 2006. <http://www.ietf.org/rfc/rfc4252.txt> [accessed 12/23/14].
- [RFC 4253] T. Ylonen and C. Lonvick (ed.), *The Secure Shell (SSH) Transport Layer Protocol*, Internet Engineering Task Force (IETF) RFC 4253, January 2006. <http://www.ietf.org/rfc/rfc4253.txt> [accessed 12/23/14].
- [RFC 4254] T. Ylonen and C. Lonvick, *The Secure Shell (SSH) Connection Protocol*, Internet Engineering Task Force (IETF) RFC 4254, January 2006. <http://www.ietf.org/rfc/rfc4254.txt> [accessed 12/23/14].
- [RFC 4255] J. Schlyter and W. Griffin, *Using DNS to Securely Publish Secure Shell (SSH) Key Fingerprints*, Internet Engineering Task Force (IETF) RFC 4255, January 2006. <http://www.ietf.org/rfc/rfc4255.txt> [accessed 12/23/14].
- [RFC 4288] N. Freed and J. Klensin, *Media Type Specifications and Registration Procedures*, Internet Engineering Task Force (IETF) RFC 4288, December 2005. <http://www.ietf.org/rfc/rfc4288.txt> [accessed 12/23/14].
- [RFC 4289] N. Freed and J. Klensin, *Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures*, Internet Engineering Task Force (IETF) RFC 4289, December 2005. <http://www.ietf.org/rfc/rfc4289.txt> [accessed 12/23/14].

- [RFC 4301] S. Kent and K. Seo, *Security Architecture for the Internet Protocol*, Internet Engineering Task Force (IETF) RFC 4301, December 2005. <http://www.ietf.org/rfc/rfc4301.txt> [accessed 12/23/14].
- [RFC 4302] S. Kent, *IP Authentication Header*, Internet Engineering Task Force (IETF) RFC 4302, December 2005. <http://www.ietf.org/rfc/rfc4302.txt> [accessed 12/23/14].
- [RFC 4303] S. Kent, *IP Encapsulating Security Payload (ESP)*, Internet Engineering Task Force (IETF) RFC 4303, December 2005. <http://www.ietf.org/rfc/rfc4303.txt> [accessed 12/23/14].
- [RFC 4307] J. Schiller, *Cryptographic Algorithms for Use in the Internet Key Exchange Version 2 (IKEv2)*, Internet Engineering Task Force (IETF) RFC 4307, December 2005. <http://www.ietf.org/rfc/rfc4307.txt> [accessed 12/23/14].
- [RFC 4308] P. Hoffman, *Cryptographic Suites for IPsec*, Internet Engineering Task Force (IETF) RFC 4308, December 2005. <http://www.ietf.org/rfc/rfc4308.txt> [accessed 12/23/14].
- [RFC 4309] R. Housley, *Using Advanced Encryption Standard (AES) CCM Mode with IPsec Encapsulating Security Payload (ESP)*, Internet Engineering Task Force (IETF) RFC 4309, December 2005. <http://www.ietf.org/rfc/rfc4309.txt> [accessed 12/23/14].
- [RFC 4346] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.1*, Internet Engineering Task Force (IETF) RFC 4346, April 2006 (obsoleted by RFC 5246). <http://www.ietf.org/rfc/rfc4346.txt> [accessed 12/23/14].
- [RFC 4419] M. Friedl, N. Provos and W. Simpson, *Diffie-Hellman Group Exchange for the Secure Shell (SSH) Transport Layer Protocol*, Internet Engineering Task Force (IETF) RFC 4419, March 2006. <http://www.ietf.org/rfc/rfc4419.txt> [accessed 12/23/14].
- [RFC 4434] P. Hoffman, *The AES-XCBC-PRF-128 Algorithm for the Internet Key Exchange Protocol (IKE)*, Internet Engineering Task Force (IETF) RFC 4434, February 2006. <http://www.ietf.org/rfc/rfc4434.txt> [accessed 12/23/14].
- [RFC 4462] J. Hutzelman, J. Salowey, J. Galbraith and V. Welch, *Generic Security Service Application Program Interface (GSS-API) Authentication and Key Exchange for the Secure Shell (SSH) Protocol*, Internet Engineering Task Force (IETF) RFC 4462, May 2006. <http://www.ietf.org/rfc/rfc4462.txt> [accessed 12/23/14].
- [RFC 4511] J. Sermersheim (ed.), *Lightweight Directory Access Protocol (LDAP) : The Protocol*, Internet Engineering Task Force (IETF) RFC 4511, June 2006. <http://www.ietf.org/rfc/rfc4511.txt> [accessed 12/23/14].
- [RFC 4512] K. Zeilenga, *Lightweight Directory Access Protocol (LDAP) : Directory Information Models*, Internet Engineering Task Force (IETF) RFC 4512, June 2006. <http://www.ietf.org/rfc/rfc4512.txt> [accessed 12/23/14].
- [RFC 4543] D. McGrew and J. Viega, *The Use of Galois Message Authentication Code (GMAC) in IPsec ESP and AH*, Internet Engineering Task Force (IETF) RFC

- 4543, May 2006. <http://www.ietf.org/rfc/rfc4543.txt> [accessed 12/23/14].
- [RFC 4556] L. Zhu and B. Tung, *Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)*, Internet Engineering Task Force (IETF) RFC 4556, June 2006. <http://www.ietf.org/rfc/rfc4556.txt> [accessed 12/23/14].
- [RFC 4635] D. Eastlake III, *HMAC SHA TSIG Algorithm Identifiers*, Internet Engineering Task Force (IETF) RFC 4635, August 2006. <http://www.ietf.org/rfc/rfc4635.txt> [accessed 12/23/14].
- [RFC 4754] D. Fu and J. Solinas, *IKE and IKEv2 Authentication Using the Elliptic Curve Digital Signature Algorithm (ECDSA)*, Internet Engineering Task Force (IETF) RFC 4754, January 2007. <http://www.ietf.org/rfc/rfc4754.txt> [accessed 12/23/14].
- [RFC 4835] V. Manral, *Cryptographic Algorithm Implementation Requirements for Encapsulating Security Payload (ESP) and Authentication Header (AH)*, Internet Engineering Task Force (IETF) RFC 4835, April 2007. <http://www.ietf.org/rfc/rfc4835.txt> [accessed 12/23/14].
- [RFC 4868] S. Kelly and S. Frankel, *Using HMAC-SHA-256, HMAC-SHA-384, and HMAC-SHA-512 with IPsec*, Internet Engineering Task Force (IETF) RFC 4868, May 2007. <http://www.ietf.org/rfc/rfc4868.txt> [accessed 12/23/14].
- [RFC 5019] A. Deacon and R. Hurst, *The Lightweight Online Certificate Status Protocol (OCSP) Profile for High-Volume Environments*, Internet Engineering Task Force (IETF) RFC 5019, September 2007. <http://www.ietf.org/rfc/rfc5019.txt> [accessed 12/23/14].
- [RFC 5035] P. Hoffman, *Enhanced Security Services (ESS) Update: Adding CertID Algorithm Agility*, Internet Engineering Task Force (IETF) RFC 5035, August 2007. <http://www.ietf.org/rfc/rfc5035.txt> [accessed 12/23/14].
- [RFC 5114] M. Lepinski and S. Kent, *Additional Diffie-Hellman Groups for Use with IETF Standards*, Internet Engineering Task Force (IETF) RFC 5114, January 2008. <http://www.ietf.org/rfc/rfc5114.txt> [accessed 12/23/14].
- [RFC 5155] B. Laurie, G. Sisson, R. Arends and D. Blacka. *DNS Security (DNSSEC) Hashed Authenticated Denial of Existence*, Internet Engineering Task Force (IETF) RFC 5155, March 2008. <http://www.ietf.org/rfc/rfc5155.txt> [accessed 12/23/14].
- [RFC 5246] T. Dierks and E. Rescorla, *The Transport Layer Security (TLS) Protocol Version 1.2*, Internet Engineering Task Force (IETF) RFC 5246, August 2008. <http://www.ietf.org/rfc/rfc5246.txt> [accessed 12/23/14].
- [RFC 5272] J. Schaad and M. Myers, *Certificate Management over CMS (CMC)*, Internet Engineering Task Force (IETF) RFC 5272, June 2008. <http://www.ietf.org/rfc/rfc5272.txt> [accessed 12/23/14].
- [RFC 5280] D. Cooper, S. Santesson, S. Farrell, S. Boeyen, R. Housley, and W. Polk, *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL)*

- Profile*, Internet Engineering Task Force (IETF) RFC 5280, May 2008.
<http://www.ietf.org/rfc/rfc5280.txt> [accessed 12/23/14].
- [RFC 5349] L. Zhu, K. Jaganathan and K. Lauter, *Elliptic Curve Cryptography (ECC) Support for Public Key Cryptography for Initial Authentication in Kerberos (PKINIT)*, Internet Engineering Task Force (IETF) RFC 5349, September 2009.
<http://www.ietf.org/rfc/rfc5349.txt> [accessed 12/23/14].
- [RFC 5430] M. Salter, E. Rescorla and R. Housley, *Suite B Profile Transport Layer Security (TLS)*, Internet Engineering Task Force (IETF) RFC 5430, March 2009.
<http://www.ietf.org/rfc/rfc5430.txt> [accessed 12/23/14].
- [RFC 5647] K. Igoe and J. Solinas, *AES Galois Counter Mode for the Secure Shell Transport Layer Protocol*, Internet Engineering Task Force (IETF) RFC 5647, August, 2009.
<http://www.ietf.org/rfc/rfc5647.txt> [accessed 12/23/14].
- [RFC 5652] R. Housley, *Cryptographic Message Syntax (CMS)*, Internet Engineering Task Force (IETF) RFC 5652, September 2009. <http://www.ietf.org/rfc/rfc5652.txt> [accessed 12/23/14].
- [RFC 5656] D. Stebila and J. Green, *Elliptic Curve Algorithm Integration in the Secure Shell Transport Layer*, Internet Engineering Task Force (IETF) RFC 5656, December 2009. <http://www.ietf.org/rfc/rfc5656.txt> [accessed 12/23/14].
- [RFC 5751] B. Ramsdell and S. Turner, *Secure/Multipurpose Internet Mail Extensions (S/MIME) Version 3.2 Message Specification*, Internet Engineering Task Force (IETF) RFC 5751, January 2010. <http://www.ietf.org/rfc/rfc5751.txt> [accessed 12/23/14].
- [RFC 5903] D. Fu and J. Solinas, *Elliptic Curve Groups modulo a Prime (ECP Groups) for IKE and IKEv2*, Internet Engineering Task Force (IETF) RFC 5903, June 2010.
<http://www.ietf.org/rfc/rfc5903.txt> [accessed 12/23/14].
- [RFC 5996] C. Kaufman, P. Hoffman, Y. Nir and P. Eronen, *Internet Key Exchange Protocol Version 2 (IKEv2)*, Internet Engineering Task Force (IETF) RFC 5996, September 2010. <http://www.ietf.org/rfc/rfc5996.txt> [accessed 12/23/14].
- [RFC 6113] S. Hartman and L. Zhu, *A Generalized Framework for Kerberos Pre-Authentication*, Internet Engineering Task Force (IETF) RFC 6113, April 2011.
<http://www.ietf.org/rfc/rfc6113.txt> [accessed 12/23/14].
- [RFC 6187] K. Igoe and D. Stebila, *X.509v3 Certificates for Secure Shell Authentication*, Internet Engineering Task Force (IETF) RFC 6187, March 2011.
<http://www.ietf.org/rfc/rfc6187.txt> [accessed 12/23/14].
- [RFC 6239] K. Igoe, *Suite B Cryptographic Suites for Secure Shell (SSH)*, Internet Engineering Task Force (IETF) RFC 6239, May 2011. <http://www.ietf.org/rfc/rfc6239.txt> [accessed 12/23/14].
- [RFC 6251] S. Josefsson, *Using Kerberos Version 5 over the Transport Layer Security (TLS)*

- Protocol*, Internet Engineering Task Force (IETF) RFC 6251, May 2011.
<http://www.ietf.org/rfc/rfc6251.txt> [accessed 12/23/14].
- [RFC 6318] R. Housley and J. Solinas, *Suite B in Secure/Multipurpose Internet Mail Extensions (S/MIME)*, Internet Engineering Task Force (IETF) RFC 6318, June 2011.
<http://www.ietf.org/rfc/rfc6318.txt> [accessed 12/23/14].
- [RFC 6379] L. Law and J. Solinas, *Suite B Cryptographic Suites for IPsec*, Internet Engineering Task Force (IETF) RFC 6379, October 2011. <http://www.ietf.org/rfc/rfc6379.txt> [accessed 12/23/14].
- [RFC 6605] P. Hoffman and W.C.A. Wijngaards, *Elliptic Curve Digital Signature Algorithm (DSA) for DNNSEC*, Internet Engineering Task Force (IETF) RFC 6605, April 2012. <http://www.ietf.org/rfc/rfc6605.txt> [accessed 12/23/14].
- [RFC 6649] L. H. Astrand and T. Yu, *Deprecate DES, RC4-HMAC-EXP, and Other Weak Cryptographic Algorithms in Kerberos*, Internet Engineering Task Force (IETF) RFC 6649, July 2012. <http://www.ietf.org/rfc/rfc6649.txt> [accessed 12/23/14].
- [RFC 6944] S. Rose, *Applicability Statement: DNS Security (DNSSEC) DNSKEY Algorithm Implementation Status*, Internet Engineering Task Force (IETF) RFC 6944, April 2013. <http://www.ietf.org/rfc/rfc6944.txt> [accessed 12/23/14].
- [RFC 6960] S. Santesson, M. Myers, R. Ankney, A. Malpani, S. Galperin and C. Adams, *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol – OCSP*, Internet Engineering Task Force (IETF) RFC 6960, June 2013.
<http://www.ietf.org/rfc/rfc6960.txt> [accessed 12/23/14].
- [RFC 7030] M. Pritikin, P. Yee and D. Harkins (eds.), *Enrollment over Secure Transport*, Internet Engineering Task Force (IETF) RFC 7030, October 2013.
<http://www.ietf.org/rfc/rfc7030.txt> [accessed 12/23/14].
- [RFC 7321] D. McGrew and P. Hoffman, *Cryptographic Algorithm Implementation Requirements and Usage Guidance for Encapsulating Security Payload (ESP) and Authentication Header (AH)*, Internet Engineering Task Force (IETF) RFC 7321, August 2014.
<http://www.ietf.org/rfc/rfc7321.txt> [accessed 12/23/14].
- [SEC1] Certicom Research, *SECI: Elliptic Curve Cryptography*, Standards for Efficient Cryptography Group, version 2.0, May 21, 2009. <http://www.secg.org/sec1-v2.pdf> [accessed 12/23/14].
- [SP 500-267] D. Montgomery, S. Nightingale, S. Frankel and M. Carson, *A Profile for IPv6 in the U.S. Government - Version 1.0*, National Institute of Standards and Technology, NIST Special Publication 500-267, July 2008, 84 pp. http://www.nist.gov/manuscript-publication-search.cfm?pub_id=900150 [accessed 12/23/14].
- [SP 800-32] D. R. Kuhn, V. C. Hu, W. T. Polk and S.-j. Chang, *Introduction to Public Key Technology and the Federal PKI Infrastructure*, National Institute of Standards and Technology, NIST Special Publication 800-32, February 26, 2001, 54 pp.

- <http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf> [accessed 12/23/14].
- [SP 800-38A] M. Dworkin, *Recommendations for Block Cipher Modes of Operation: Methods and Techniques*, National Institute of Standards and Technology, NIST Special Publication 800-38A 2001 Edition, December 2001, 66 pp.
<http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf> [accessed 12/23/14].
- [SP 800-49] C. M. Chernick, *Federal S/MIME V3 Client Profile*, National Institute of Standards and Technology, NIST Special Publication 800-49, November 2002, 27 pp.
<http://csrc.nist.gov/publications/nistpubs/800-49/sp800-49.pdf> [accessed
- [SP 800-52] T. Polk, K. McKay and S. Chokhani, *Guidelines for the Selection, Configuration and Use of Transport Layer Security (TLS) Implementations*, National Institute of Standards and Technology, NIST Special Publication 800-52 Revision 1, April 2014, 67 pp. <http://dx.doi.org/10.6028/NIST.SP.800-52r1>.
- [SP 800-56A] E. Barker, L. Chen, A. Roginsky and M. Smid, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, National Institute of Standards and Technology, NIST Special Publication 800-56A Revision 2, May 2013, 138 pp. <http://dx.doi.org/10.6028/NIST.SP.800-56Ar2>.
- [SP 800-56B] E. Barker, L. Chen and D. Moody, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*, National Institute of Standards and Technology, NIST Special Publication 800-56B Revision 1, September 2014, 131 pp. <http://dx.doi.org/10.6028/NIST.SP.800-56Br1>.
- [SP 800-57 Part 1] E. Barker, W. Barker, W. Burr, W. Polk and M. Smid, *Recommendation for Key Management – Part 1: General*, National Institute of Standards and Technology, NIST Special Publication 800-57 Part 1 Revision 3, July 2012, 147 pp.
http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf
[accessed 12/23/14].
- [SP 800-67] W. C. Barker and E. Barker, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, National Institute of Standards and Technology, NIST Special Publication 800-67 Revision 1, January 2012, 34 pp.
<http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf> [accessed 12/23/14].
- [SP 800-77] S. Frankel, K. Kent, R. Lewkowski, A. D. Orebaugh, R. W. Ritchey and S. R. Shama, *Guide to IPsec VPNs*, National Institute of Standards and Technology, NIST Special Publication 800-77, December 2005, 126 pp.
<http://csrc.nist.gov/publications/nistpubs/800-77/sp800-77.pdf> [accessed 12/23/14].
- [SP 800-81] R. Chandramouli and S. Rose, *Secure Domain Name System (DNS) Deployment Guide*, National Institute of Standards and Technology, NIST Special Publication 800-81-2, September 2013, 130 pp. <http://dx.doi.org/10.6028/NIST.SP.800-81-2>.
- [SP 800-89] E. Barker, *Recommendation for Obtaining Assurances for Digital Signature*

Applications, National Institute of Standards and Technology, NIST Special Publication 800-89, November 2006, 38 pp.

http://csrc.nist.gov/publications/nistpubs/800-89/SP-800-89_November2006.pdf
[accessed 12/23/14].

- [SP 800-90A] E. Barker and J. Kelsey, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, National Institute of Standards and Technology, NIST Special Publication (DRAFT) 800-90A, November 2014, 112 pp. <http://csrc.nist.gov/publications/PubsSPs.html> [accessed 12/23/14].
- [SP 800-90B] E. Barker and J. Kelsey, *Recommendation for the Entropy Sources Used for Random Bit Generation*, National Institute of Standards and Technology, NIST Special Publication (DRAFT) 800-90B, August 2012 [September 2013], 78 pp. <http://csrc.nist.gov/publications/PubsSPs.html> [accessed 12/23/14].
- [SP 800-90C] E. Barker and J. Kelsey, *Recommendation for Random Bit Generator (RBG) Constructions*, National Institute of Standards and Technology, NIST Special Publication (DRAFT) 800-90C, August 2012 [September 2013], 50 pp. <http://csrc.nist.gov/publications/PubsSPs.html> [accessed 12/23/14].
- [SP 800-118] K. Scarfone and M. Souppaya, *Guide to Enterprise Password Management*, National Institute of Standards and Technology, NIST Special Publication (DRAFT) 800-118, April 2009, 38 pp. <http://csrc.nist.gov/publications/PubsSPs.html> [accessed 12/23/14].
- [SP 800-131A] E. Barker and A. Roginsky, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, National Institute of Standards and Technology, NIST Special Publication 800-131A, January 2011, 27 pp. <http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf> [accessed 12/23/14].
- [SP 800-132] M. S. Turan, E. Barker, W. Burr and L. Chen, *Recommendation for Password-Based Key Derivation, Part 1: Storage Applications*, National Institute of Standards and Technology, NIST Special Publication 800-132, December 2010, 18 pp. <http://csrc.nist.gov/publications/nistpubs/800-132/nist-sp800-132.pdf> [accessed 12/23/14].
- [SP 800-135] Q. Dang, *Recommendation for Existing Application-Specific Key Derivation Functions*, National Institute of Standards and Technology, NIST Special Publication 800-135 Revision 1, December 2011, 23 pp. <http://csrc.nist.gov/publications/nistpubs/800-135-rev1/sp800-135-rev1.pdf> [accessed 12/23/14].
- [TPM] Trusted Computing Group, *TPM Main Specification Version 1.2, Revision 103 [Parts 1 (Design Principles) , 2 (TPM Structures) and 3 (Commands)]*, July 9, 2007. http://www.trustedcomputinggroup.org/resources/tpm_main_specification [accessed

12/23/14].

[X9.62] American National Standards Institute, *Public Key Cryptography for the Financial Services Industry: the Elliptic Curve Digital Signature Algorithm (ECDSA)* , ANSI X9.62:2005, November 2005.

[X11] X.Org Foundation, *Documentation for the X Window System Version 11 Release 7.6 (X11R7.6)* , December 2010. <http://www.x.org/releases/X11R7.6/doc/index.html> [accessed 12/23/14].

[X.509 Path] D. Cooper and W. T. Polk, *NIST Recommendation for X.509 Path Validation, Version 0.5*, Draft Special Publication 800-XXX, May 3, 2004, 28 pp. http://csrc.nist.gov/groups/ST/crypto_apps_infra/documents/NIST_Recommendation_for_X509_PVMs.pdf [accessed 12/17/14].

附属書 E：改訂履歴

本改訂版は、[SP 800-131A]で規定されるとおり、必要なセキュリティ強度が達成されることが可能となるように、本書におけるプロトコルおよびアプリケーションのための暗号学的な要求事項についてアップデートする。本改訂版は、本書において対処されるプロトコルのセキュリティ関連のアップデートについても追加する。さらに、本改訂版はセキュアシェル（SSH）のための新しいセクションを追加する。本改訂版における具体的な変更は以下を含む。

セクション 2：

- 1) デジタル署名と鍵確立証明書についての推奨されるアルゴリズムと鍵サイズについてのさらなる明確化が追加された。
- 2) 2013年以降は、1024-bit RSAとDSAは、もはやデジタル署名生成に関して承認されない。
- 3) CA/RAソフトウェアとハードウェア（[セクション 2.3.1](#)）が、LDAPのサポートのため、現在は推奨（“すべきである”を用いて）となった、初期のバージョンでの必要（“しなければならない”を用いて）ではなくなった。
- 4) 依拠当事者のソフトウェア（[セクション 2.3.5](#)）は、証明書状態のためにOCSPまたはCRLのいずれかをサポートすることが必要である。本書の第一版では、CRLのサポートが必須であった。
- 5) 必須の“anyExtendedKeyUsage”機能のサポートが削除された。

セクション 3:

- 1) VPN暗号スイートの情報がスイートB VPN暗号スイートを含むようにアップデートされた。
- 2) 表 3-2および暗号アルゴリズム推奨が改訂された。

セクション 4:

- 1) TLSセクションの改訂が将来SP 800-52 改訂第一版[SP 800-52]であるだろう。

セクション 5:

- 1) 暗号スイート（ALS）の議論がアップデートされた。これら暗号スイートの使用についての新しい要求事項が[SP 800-131A]で規定されるセキュリティ要求事項を満たすために追加された。
- 2) [セクション5.3](#)の項目3で、“すべきである”が“してもよい”に変更された。

セクション 6:

- 1) 2013年以降の112 bit のセキュリティの要求事項が公開鍵認証と鍵確立について記述された。
- 2) 認証と鍵確立のためのTLS使用のサポートが追加された。

セクション 7:

[セクション 7.2.3](#)にて、本バージョンでの2つの“すべきである”が“しなければならない”に変更となった。

セクション 8:

- 1) “すべきである”の要求事項が2013年末までのSHA-256を用いた2048-bit RSAに対して減すよう追加された。
- 2) ゴーン署名のためのSHA-1を用いたRSAとDSAの承認が2013年末までに制限された；表8-1参照
- 3) メッセージ認証アルゴリズムが表8-2において改訂された。

[セクション 9:](#) 本セクションには主要な改訂はない。

[セクション 10:](#) 新しいセクション。