NIST Special Publication 800-52
Revision 1

トランスポート層セキュリティ(TLS)実装の選択、設定、および使用のためのガイドライン

Tim Polk
Kerry McKay
Santosh Chokhani

http://dx.doi.org/10.6028/NIST.SP.800-52

コンピュータ　セキュリティ

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 800-52
Revision 1

トランスポート層セキュリティ(TLS)実装の選択、設定、および使用のためのガイドライン

Tim Polk
Kerry McKay
コンピュータ セキュリティ部門
情報技術研究所

Santosh Chokhani
シグナコム ソリューションズ
McLean, VA

http://dx.doi.org/10.6028/NIST.SP.800-52

2014 年 4 月

米国商務省
Penny Pritzker　長官

米国国立標準技術研究所
Patrick D. Gallagher　標準技術担当次官兼所長

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 800-52 Revision 1
Natl. Inst. Stand. Technol. Spec. Publ. 800-52 Revision 1, 66 pages (April 2014)
CODEN: NSPUE2

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。

本文書についてのコメントは以下へ提出してください：

National Institute of Standards and Technology
Attn: Computer Security Division, Information Technology Laboratory
100 Bureau Drive (Mail Stop 8930) Gaithersburg, MD 20899-8930
Email: SP80052-comments@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、管理面およ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

トランスポート層セキュリティ(TLS)は、インターネット上の電子的な情報提供中の機微な情報を保護するためのメカニズムを提供する。この Special Publication は、連邦政府情報処理規格(FIPS)および NIST 推奨暗号アルゴリズムを効果的に使用する際に、TLS プロトコル実装の選択と設定のガイダンスを提供し、また TLS 1.1 が FIPS ベースの暗号スイートを最低限適切なセキュアなトランスポートプロトコルとなるよう設定されることを要求し、2015 年 1 月 1 日までに TLS 1.2 への移行計画を政府機関が策定することを推奨する。この Special Publication は、提供されなければならない必須のサポートの TLS 拡張およびその他の推奨される拡張についても特定する。

キーワード

情報セキュリティ；ネットワークセキュリティ；SSL；TLS；トランスポート層セキュリティ

謝辞

共著者、NIST の Tim Polk 氏および Kerry McKay 氏、および CygnaCom Solutions の Santosh Chokhani 氏は、本文書の策定にご協力いただいた数多くの人々に感謝の意を表します。特に本書初版公開バージョンの著者である、NIST の Matthew J.Fanto 氏および C. Michael Chernick 氏 および Booz Allen and Hamilton 社の Charles Edington III 氏および Rob Rosenthal 氏に深く感謝の意を表します。

目次

Executive Summary vii

1 序説 1

1.1 背景 1
1.2 TLS の歴史 1
1.3 適用範囲 2
1.4 文書の表記 3

2 TLS 概要 4

2.1 ハンドシェイクプロトコル 4
2.2 共有秘密ネゴシエーション 5
2.3 機密性 6
2.4 完全性 6
2.5 認証 6
2.6 耐リプレイ 7
2.7 鍵管理 7

3 TLS サーバの最小限の要求事項 9

3.1 プロトコルバージョンサポート 9
3.2 サーバ鍵と証明書 9
3.2.1 サーバ証明書プロファイル 10
3.2.2 クライアント証明書の失効状態情報の取得 13
3.2.3 サーバ公開鍵証明書の保証 14
3.3 暗号サポート 14
3.3.1 暗号スイート 15
3.3.2 認証された暗号 19
3.4 TLS 拡張サポート 20
3.4.1 必須の TLS 拡張 20
3.4.2 条件付き TLS 拡張 21
3.4.3 推奨されない TLS 拡張 22
3.5 クライアント認証 23
3.5.1 パス検証 23
3.5.2 トラストアンカストア 24
3.5.3 クライアント鍵長のチェック 24
3.5.4 サーバヒントリスト 24

3.6 セッション再開 25
3.7 圧縮方法 25
3.8 運用上の検討事項 25
3.9 サーバ推奨事項 26
3.9.1 サーバ選択の推奨事項 26
3.9.2 サーバのインストールと設定のための推奨事項 27
3.9.3 サーバシステム管理者のための推奨事項 30

4 TLS クライアントの最小限の要求事項 31

4.1 プロトコルバージョンサポート 31
4.2 クライアント鍵と証明書 31
4.2.1 クライアント証明書プロファイル 31
4.2.2 サーバ証明書の失効状態情報の取得 34
4.2.3 クライアント公開鍵証明書の保証 35
4.3 暗号サポート 35
4.3.1 暗号スイート 35
4.3.2 認証された暗号 35
4.4 TLS 拡張サポート 35
4.4.1 必須の TLS 拡張 35
4.4.2 条件付き TLS 拡張 36
4.4.3 推奨されない TLS 拡張 37
4.5 サーバ認証 37
4.5.1 パス検証 37
4.5.2 トラストアンカストア 38
4.5.3 サーバ鍵長のチェック 39
4.5.4 利用者インタフェース 39
4.6 セッション再開 39
4.7 圧縮方法 39
4.8 運用上の検討事項 40
4.9 クライアント推奨事項 40
4.9.1 クライアント選択の推奨事項 40
4.9.2 クライアントのインストールと設定のための推奨事項 41
4.9.3 クライアントシステム管理者のための推奨事項 43
4.9.4 エンドユーザのための推奨事項 44

附属書 A：略語 45

附属書 B：暗号スイート名の解釈 46

附属書 C：事前共有鍵 48

附属書 D：将来の機能 50

D.1　追加の／代替のウェブサーバ証明書検証メカニズム 50
D.1.1　ソブリンキー（Sovereign Keys） 50
D.1.2　証明書の透明性 50
D.1.3　パースペクティブズ（Perspectives）とコンバージェンス（Convergence） 50
D.1.4　DANE 51
D.2　サーバ／クライアント鍵長のチェック 52
D.3　Encrypt-then-MAC 拡張 52

附属書 E　参考文献 53

Executive Summary

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、政府機関の情報資源の管理（Management of Federal Information Resources）は、機微な情報であるが非格付け(訳注：機密)データを含むような公的アクセス可能な情報リポジトリまたは情報提供システムの管理者に対して、機微なデータが喪失、誤使用または許可されないアクセス、またはこのようなデータの改ざんから帰結するリスクおよび損害の重要さと同一程度の保護が為されることを保証するよう要求する。情報を共有する相互接続ネットワークの性質およびインターネットの使用を前提とし、この機微なデータの保護は、そのデータを保護するために適切なメカニズムが採用されない場合には困難となる可能性がある。トランスポート層セキュリティ(TLS)は、インターネット上の電子的な情報提供中に機微なデータを保護するためのメカニズムを提供する。

TLS は、二つの通信アプリケーションの間の認証、機密性、およびデータ完全性を提供するために作成されたプロトコルである。TLS は、前身のセキュアソケットレイヤーバージョン 3.0 (SSL 3.0) と呼ばれるプロトコルに基づいており、SSL 3.0 に対する改良版であると考えられる。SSL 3.0 は、[RFC 6101] で規定される。トランスポート層セキュリティバージョン 1 (TLS 1.0) 仕様は、インターネット Request for Comments [RFC 2246]で規定される。それぞれの文書は、インターネット上のセキュリティサービスを提供するような類似のプロトコルを規定する。TLS 1.0 は[RFC 4346]で記述されるとおり、バージョン 1.1 へ改訂され、TLS 1.1 はさらに[RFC 5246]で記述されるとおり、バージョン 1.2 へ改訂された。さらにいくつかの拡張が TLS を用いた実装における既知のセキュリティ脆弱性のいくつかを軽減するために定義されている。これらの脆弱性は必ずしも TLS での弱点ではないが、TLS をアプリケーションがどう使うかに関係する。

この Special Publication は、承認された暗号スキームとアルゴリズムを効果的に使用する際の TLS プロトコル実装の選択および設定に対するガイダンスを提供する。本書は、特に TLS 1.1 が最低限適切なセキュアトランスポートプロトコル [1]として、承認されたスキームとアルゴリズムを用いた暗号スイートを設定されることを要求する。2015 年 1 月 1 日までに、承認されたスキームとアルゴリズムを用いて設定された TLS 1.2 への移行計画を政府機関が策定することについても推奨する。非政府システムとの相互運用性が要求されるとき、TLS 1.0 がサポートされてもよい。この Special Publication は、提供されなければならない必須のサポートの TLS 拡張およびその他の推奨される拡張についても特定する。

この Special Publication で提供される推奨事項の使用は、以下を促進するだろう：

- インターネット上の情報配送の保護のための、認証、機密性および完全性メカニズムのより一貫した使用；
- NIST 承認されたアルゴリズムおよび公開標準を含む推奨暗号スイートの一貫した使用；
- TLS プロトコル上の既知および想定される攻撃に対する保護；および
- トランスポート層セキュリティ実装の統合におけるシステム管理者や管理者（マネージャー）による十分な情報を得た上での決定。

これらのガイドラインは主に連邦政府利用者およびシステム管理者が、機微ではあるが機密ではないような米国連邦政府のデータをインターネット上の重大な脅威から適切に保護するよう設計されているが、それらは、閉鎖されたネットワーク環境内でデータを隔離するために使用されてもよい。(議論され

[1] SSL 3.0 は、SLL プロトコルバージョンの最もセキュアなものであるが、連邦政府情報の保護では使用が承認されていない、なぜなら承認されていない暗号アルゴリズムの使用に一部依存しているからである。TLS バージョン 1.1 および 1.2 は、適切に構成されるとき、連邦政府情報の保護に承認されている。TLS 1.0 は、非政府システムと相互運用性のために必須であり、これらのガイドラインにしたがって構成されるときにのみ承認される。

るクライアント-サーバモデルとセキュリティサービスはこれらの状況でも適用する。）この Special Publication は、NIST Special Publication 800-52 に優先する。本 Special Publication は既存の方針と手順と共に使用されるべきである。

1 序説

多くのネットワークアプリケーションは、セキュアでないチャネルを介して送信される機密データを保護するために、セキュアソケットレイヤー(SSL)とトランスポート層セキュリティ(TLS)プロトコルに依存する。インターネットのクライアント-サーバ モデルと通信プロトコルの設計原則は、[Rescorla01]、[Comer00]、および[Hall00]のような、多くの書籍で記述されている。TLS は、[RFC 5280]に準拠した公開鍵証明書を生成する公開鍵基盤(PKI)の存在を必要とする。[Adams99]や[Housley01]のような書籍は、技術ジャーナル記事(例. [Polk03])や NIST 公開文書(例. [SP800-32])と同様に、インターネット上の情報を保護するために PKI を使用する方法について記述する。

本書は、これらのガイドラインの読者が、例えば、X.509 証明書；および SSL/TLS プロトコルを含め、公開鍵基盤の概念に精通していると想定する。上記および附属書 E で述べた参考文献はこれらのガイドラインで十分に説明されないような背景をさらに説明する。

1.1 背景

TLS プロトコルは、さまざまなオンライントランザクションの通信をセキュアにするために使用される。このようなトランザクションには、金融トランザクション（例. 銀行、株式取引、電子商取引)、健康管理(例. 医療記録閲覧や医療予約スケジュール)、および社会的トランザクション(例. 電子メールやソーシャルネットワーク）が含まれる。個人識別情報（PII：Personally Identifiable Information)、金融データまたはログイン情報など、機微なデータや価値のあるデータを取り扱うネットワークサービスは、そのデータを適切に保護する必要がある。TLS はサーバとクライアント間でデータを送信するための保護されたチャネルを提供する。クライアントは、必ずしもそうではないが、大抵はウェブブラウザである。

TLS は、信頼の高いトランスポートプロトコル － 通常はトランスミッションコントロールプロトコル(TCP) － の上で動作する階層化プロトコルである。ハイパーテキスト転送プロトコル(HTTP)やインターネットメッセージアクセスプロトコル(IMAP)のようなアプリケーションプロトコルは、TLS 上で動作可能である。TLS は、アプリケーションに依存せず、アプリケーションプロトコル経由でネットワークを介してデータを送信する二つの通信アプリケーションに対してセキュリティを提供するために使用される。外部システムを内部ネットワークへ接続するような仮想プライベートネットワーク(VPN)を生成し、そのシステムがネットワーク内にあるかのように多数の内部サービスとリソースにアクセスできるようにすることが可能である。

1.2 TLS の歴史

SSL プロトコルは、クライアントサーバアプリケーションのセキュリティニーズを満たすため Netscape Corporation[2]によって設計された。SSL のバージョン 1 は、リリースされなかった。SSL2.0 が 1995 年にリリースされたが、良く知られるセキュリティ脆弱性を持っていたため、1996 年の SSL 3.0 のリリースによって対処された。この期間中、Microsoft Corporation は、プライベートコミュニケーションテクノロジー(PCT)として知られるプロトコルをリリースし、その後、パフォーマンスの高い、セキュアトランスポートレイヤープロトコル(STLP)として知られるプロトコルをリリースした。

PCT と STLP は SSL 2.0 と SSL 3.0 が支配する市場シェアを占有することはなかった。異なる実装をの間で通信の互換性を保証するためのインターネット標準の開発に責任を持つ技術ワーキンググループである Internet Engineering Task Force (IETF)は、プロトコル間のセキュリティエンジニアリングとプロトコルの非互換性の問題についてできるだけの解決を試みた。IETF 標準トラックであるトランスポート層セキュリティプロトコルバージョン 1.0 (TLS 1.0)が登場し、IETF によって[RFC2246]として成文化された。TLS 1.0 は、SSL 3.0 に基づいており、それらの違いは劇的ではないが、TLS 1.0 と

[2] 民間の会社名が歴史的参照目的でのみ使用される。一切の製品の是認は意図されていない、または含まれない。

SSL 3.0 は相互接続しないことは十分な意義を持っている。TLS 1.0 は SSL 3.1 としても参照される。

TLS 1.0 は、TLS1.0 実装があたかも TLS が提案されなかったかのように、SSL3.0 を用いて要求するエンティティとネゴシエートできるようなメカニズムを持っている。しかし、SSL 3.0 は連邦政府情報の保護(セクション D.9 の[FIPS140Impl])において使用することが承認されていないため、連邦政府情報が保護されるときに SSL 3.0 のネゴシエーションと使用が決して発生しないように、TLS は適切に設定されなければならない。

TLS 1.1 は、主として、初期化ベクタ選択とパディングエラー処理において TLS 1.0 で発見された弱点に対処するために開発された。初期化ベクタは TLS によって使用される Cipher Block Chaining(CBC)利用モード上の特定のクラスの攻撃を防止するために明示的に [3]された。パディングエラーの取扱いは、パディングエラーを復号失敗ではなく、バッドメッセージ認証コードとして扱うよう変更された。さらに、TLS 1.1 RFC は、メッセージ認証コード(MAC)を計算するために時刻に信頼を置くような CBC モード上の攻撃を知らせる。[RFC4346]は、このような攻撃に対して防御するため、実装はパディングエラーが存在するかどうかに拘わらず同じやり方でレコードを処理しなければならないと述べている。さらに[RFC4346]に含まれない CBC モードの実装に関する検討事項はセクション 3.3.1.1 で議論される。

TLS 1.2 は、特にハッシュ関数の領域で、ハッシュ、MAC および疑似乱数関数(PRF)の計算に SHA-2 ファミリアルゴリズムを使用または指定できるなど、いくつかの暗号学的な強化がなされた。TLS 1.2 は、暗号化と認証を同時に行う認証付き暗号(AEAD)暗号スイートのサポートも追加した。

1.3 適用範囲

セキュリティは、一つのプロトコルの持つ一つの特性ではない。むしろセキュリティには、要求される情報保証の特徴と情報保護サービスを共に提供するような関連する一連の複雑な特性を含む。セキュリティ要求事項は、通常、脅威または敵対者がシステムに対して仕掛けるような攻撃に対するリスク評価から導き出される。敵対者は、コンピュータオペレーティングシステム、アプリケーションソフトウェアシステム、およびそれらを相互接続するようなコンピュータネットワークを含め、多くのシステム設定要素で見つかる実装の脆弱性を利用すると思われる。したがって、無数の脅威に対してシステムをセキュアにするため、セキュリティはさまざまなシステムとネットワーク層において賢く配置されなければならない。

これらのガイドラインは、ネットワーク内のセキュリティ上にのみ焦点を当て、トランスポート層として参照されるようなネットワーク通信スタックの小さな部分に直接焦点を当てる。いくつかのその他の NIST 公開文書はシステムおよびネットワーク層のその他の部分におけるセキュリティ要求事項に対処している。本ガイドラインは、通信データのみを保護する。その他の適用可能な NIST 標準とガイドラインは、システムと保存データの保護を保証するために使用されるべきである。

これらのガイドラインは、クライアントとサーバが幅広い種類の実装と相互運用しなければならず、また公開鍵証明書を用いて認証が実行されるような、通常の用途に焦点を当てる。相互運用性を促進するため、これらのガイドライン(および TLS プロトコルを定義するような RFC)は、サポートしなければならない実装に適合するような必須の機能と暗号スイートを確立する。しかし、セキュリティが必要とされるが、幅広い相互運用性は要求されず、使用されない機能を実装するコストが禁止されるような、より多くの制限された TLS サーバの実装がある。例えば、最小限のサーバは、しばしば組み込みコントローラおよびルータのようなネットワーク基盤デバイスにおいて実装され、デバイスをリモートに設定管理するためにブラウザを使用する。これらのガイドラインで規定された機能の適切なサブセットを使用することは、このような場合には許容可能かもしれない。

TCP/IP と組み合わせて使用されるとき、TLS はさらに制限される。例えば、 Datagram TLS (DTLS)

---

[3] 初期化ベクトル(IV)は、送信されなければならない；それは、以前のメッセージ等、両方の当事者によって知られる状態から導出できない。

は、これらのガイドラインの対象外である。NIST は、DTLS の別のガイドラインを後日発行するかもしれない。

1.4 文書の表記

本文書の全体を通じて、要求事項を特定するためにキーワードが使用される。キーワード「shall」、「shall not」、「should」、および「should not」が使用される。これらの用語は、IETF Request for Comments (RFC) 2119 のキーワードであり、その他の規程文書[RFC2119]における表記法に基づいて選択されている。キーワードに追加して、用語「need」、「can」、および「may」が本文書においてスキームまたはアルゴリズムが連邦政府情報処理標準(FIPS)で記述されていることを示し、または NIST によって推奨されていることを示す。

本文書の推奨事項は、サーバ推奨事項とクライアント推奨事項にグループ化されている。セクション 3 は TLS サーバの選択と設定についての詳細なガイダンスを提供する。セクション 3.9.1 は、TLS サーバの選択に適用されるガイダンスを、セクション 3.9.2 は、TLS サーバ実装の設定に適用されるガイダンスを、セクション 3.9.3 は、サーバを維持する責任のあるシステム管理者のガイダンスを要約している。セクション 4 では、TLS クライアントの選択、設定、および使用についての詳細なガイダンスが提供される。セクション 4.9.1 は、TLS クライアント実装の選択に適用されるガイダンスを、セクション 4.9.2 は、TLS クライアント実装の設定に適用されるガイダンスを、セクション 4.9.3 は、TLS クライアントの維持に責任を持つシステム管理者のガイダンスを要約しており、セクション 4.9.4 は、エンドユーザのガイダンスを含んでいる。

2 TLS 概要

TLS は、TLS レコードプロトコル上でレコードを交換する。TLS レコードには、バージョン情報、アプリケーションプロトコルデータ、およびアプリケーションデータの処理に使用される上位レベルのプロトコルなど、いくつかのフィールドが含まれている。TLS は、機密性、完全性、および交換されるアプリケーションデータの真正性を保証するために一連の暗号アルゴリズムを用いることによってアプリケーションデータを保護する。TLS は、各プロトコルがそれ自身のレコードタイプを持つような、レコードプロトコルの最上位にあるコネクション管理のためのいくつかのプロトコルを定義する。セクション 2.1 で議論される、これらのプロトコルは、セキュリティパラメータを確立および変更し、またサーバとクライアントへのエラーおよびアラート条件を通信するために使用される。セクション 2.2 から 2.6 には、TLS プロトコルによって提供されるセキュリティサービス、およびそれらのセキュリティサービスがどのように初期設定されるかについて記述する。セクション 2.7 には鍵管理について議論する。

2.1 ハンドシェイクプロトコル

セッションコネクションを制御するために使用されるような TLS プロトコルには三つのサブプロトコルがある：ハンドシェイク、暗号スペック変更 [4]、およびアラートプロコル。TLS ハンドシェイクプロトコルは、セッションパラメータをネゴシエートするために使用される。アラートプロコルは、エラー状態の相手方に通知するために使用される。暗号スペック変更プロトコルは、あるセッションの暗号パラメータを変更するために使用される。さらに、クライアントとサーバは、ネゴシエーションされた暗号スイートによって設定されたセキュリティサービスによって保護されるようなアプリケーションデータを交換する。これらのセキュリティサービスは、ハンドシェイクを用いてネゴシエートされ、確立される。

ハンドシェイクプロトコルは、クライアントとサーバの間の一連のメッセージ交換から成る。ハンドシェイクプロトコルは、鍵確立、ディジタル署名、機密性および完全性アルゴリズムを含め、暗号スイートのアルゴリズムおよび機能をネゴシエートすることによって、オプションの暗号機能を使用するために、クライアントとサーバの両方を初期化する。クライアントとサーバは、一つまたは複数の以下のセキュリティサービスがハンドシェイク中にネゴシエートされるように設定可能である：機密性、メッセージ完全性、認証、およびリプレイ保護。機密性サービスは、データを秘密に保ち盗聴を防止するという保証を提供する。メッセージ完全性サービスは、許可されないデータ変更が検知されたことを確認し、検知されないデータの削除、追加、または改ざんを防止する。認証サービスは、送信者または受信者の同一性の保証を提供し、それによって偽造を検知する。リプレイ保護は、許可されない利用者が以前のデータを取り込めず、リプレイに成功しないことを保証する。これらのガイドラインに適合するため、クライアントとサーバの両方がデータ機密性と完全性サービスについて設定されなければならない。単純増加するシーケンス番号を含み、データ完全性が保証されるとき、耐リプレイサービスは暗黙的であることに注意すること。

ハンドシェイクプロトコルは、サーバとクライアントを相互に認証するために、オプションで X.509 公開鍵証明書 [5]を交換するために使用される。これらのガイドラインに適合するため、サーバは常にこれらのガイドラインの他の場所で記述される要求事項に適合するような X.509 公開鍵証明書を提示する。クライアント認証されたコネクションについて、クライアントもこれらのガイドラインの他の場所で記述される要求事項に適合するような X.509 公開鍵証明書を提示する

ハンドシェイクプロトコルは、セッションパラメータを確立するために責任がある。クライアントとサ

---

4 これらのガイドラインで、「change cipher spec」は、プロトコルを参照し、「ChangeCipherSpec」は、プロトコルにおいて使用されるメッセージを参照する

5 X.509 公開鍵証明書の使用は、TLS にとっての基盤である。X.509 公開鍵証明書の包括的な説明については、[Adams99]または[Housely01]を参照。これらのガイドラインでは、用語「証明書」と「公開鍵証明書」は交換できるように使用される。

ーバは、データ圧縮のように、対称鍵を導出し、その他のセッションパラメータを確立するのと共に、認証、機密性および完全性のためのアルゴリズムをネゴシエートする。ネゴシエートされた一連の認証、機密性、および完全性アルゴリズムは、暗号スイート(Cipher Suite)と呼ばれる。

すべてのセキュリティパラメータが適切であるとき、ハンドシェイク中にネゴシエートされたセキュリティサービスを開始するよう相手側へ通知するために ChangeCipherSpec メッセージが使用される。ChangeCipherSpec メッセージ後に送信されたすべてのメッセージがネゴシエートされた暗号スイートと導出された対称鍵を用いて保護される(暗号化および／または完全性保護される)。

ChangeCipherSpecメッセージの直後に送信される、Finishedメッセージは、ハンドシェイクメッセージの完全性チェックを提供する。それぞれのFinishedメッセージは、ネゴシエートされた暗号スイートと導出されたセッション鍵を用いて保護される。それぞれの側は、それらのFinishedメッセージを含まず、それまでのすべてのハンドシェイクメッセージのハッシュ値を保持する(例．サーバによって送信されるFinishedメッセージは、クライアントによって送信されたFinishedメッセージをハッシュ値に含む)。Finishedメッセージを形成するためのマスタ秘密鍵による鍵付疑似乱数関数(PRF)を通してハッシュ値は送信される。受信側は、保護されたFinishedmesse-ji を復号し、ハッシュメッセージ上のPRFの出力と比較する。PRF値が異なる場合、ハンドシェイクは改ざんされたか、鍵管理においてエラーが発生して、コネクションが中止される。PRF値が同じ場合、ハンドシェイク全体が暗号学的に完全性を持ち、何も改ざん、追加または削除されていないこと、およびすべての鍵導出が正常に行われたというより高い保証となる。

アラートが、エラーや警告等のセッションについての情報を運ぶために使用される。例えば、アラートは、復号エラー(decrypt_error) またはアクセスが拒否されたこと(access_denied)を示すために使用可能である。警告のために使用されるいくつかのアラート、およびその他は、致命的とみなされ、セッションの緊急終了へと導く。close_notifyアラートメッセージがセッションの通常終了を通知するために使用される。ハンドシェイクプロトコルが完了した後のその他すべてのメッセージのように、アラートメッセージは、暗号化され、オプションで圧縮される。

ハンドシェイク、暗号スペック変更およびアラートプロトコルは、これらのガイドラインの適用範囲外である；それらは、[RFC5246]で記述される。

2.2 共有秘密ネゴシエーション

クライアントとサーバは、TLS ハンドシェイクプロトコル中に鍵材料を確立する。プリマスタシークレットの導出は合意される鍵交換方法に依存する。例えば、リベスト・シャミア・エーデルマン(RSA)アルゴリズムが鍵交換で使用されるとき、プリマスタシークレットがクライアントによって生成され、サーバの公開鍵で暗号化されて、ClientKeyExchange メッセージでサーバに送信される。Diffie-Hellman アルゴリズムが鍵交換アルゴリズムとして使用されるとき、クライアントとサーバは相互にパラメータを送信しあい、その結果生成される鍵がプリマスタシークレットとして使用される。プリマスタシークレットは、hello メッセージでクライアントとサーバによって交換されるランダムな値と共に、マスタシークレットを計算するために使用される。セクション 2.3 および 2.4 で記述されるように、マスタシークレットは、クライアントとサーバ間で交換されるデータを保護するためのネゴシエートされたセキュリティサービスによって使用されるようなセッション鍵を導出するために使用され、従ってクライアントとサーバが通信するためのセキュアなチャネルを提供する。耐リプレイのための保護は、それぞれのパケットは単純に増加するシーケンス番号を持つため、暗黙的に提供される。

これらの秘密の確立は盗聴に対してセキュアである。TLS プロトコルがこれらのガイドラインに従って使用されるとき、秘密と同様に、アプリケーションデータはコネクションの中間にいるような攻撃者に対して脆弱ではない。攻撃者は、クライアントとサーバによって検知されることなしにハンドシェイクメッセージを改ざんすることはできない、なぜならばセキュリティパラメータ確立後に交換される Finished メッセージが交換全体にわたる完全性保護を提供するからである。言い換えれば、攻撃者は、ネゴシエーションの中間にいることによってコネクションのセキュリティを改ざんまたはダウングレ

ードすることができない。

プリマスタシークレットが RSA 鍵配送、Diffie-Hellman(DH または DHE)鍵共有、または楕円曲線 DH(ECDH または ECDHE)を用いてクライアントによってセキュアに確立される。

2.3 機密性

機密性は、暗号スイートおよびマスタシークレット、ランダム値から導出された暗号鍵、一つはクライアントによる暗号化のため(クライアント書込み鍵)、および別のものはサーバによる暗号化(サーバ書込み鍵)についてのネゴシエーションされた暗号アルゴリズムによる通信セッションのために提供される。メッセージの送信者(クライアントまたはサーバ)は導出された暗号鍵を用いてメッセージを暗号化する；受信者はそのメッセージを復号するために同じ鍵を使用する。クライアントとサーバの両方は、これらの鍵を知っており、暗号化で使用された同じ鍵を用いてメッセージを復号する。暗号化鍵は共有されるマスタシークレットから導出される。

2.4 完全性

ネゴシエートされた暗号スイートによって規定される、鍵付 MAC アルゴリズムは、メッセージの完全性を提供する。二つの MAC 鍵が導出される：1) クライアントがメッセージ送信者であり、サーバがメッセージ受信者であるときに使用されるべき MAC 鍵(クライアント書込み MAC 鍵)、および 2) サーバがメッセージ送信者であり、クライアントがメッセージ受信者であるときに使用されるべき 2 番目の MAC 鍵(サーバ書込み MAC 鍵)。メッセージの送信者(クライアントまたはサーバ)が、適切な MAC 鍵を用いてメッセージの MAC を計算し、適切な暗号鍵を用いてメッセージと MAC の両方を暗号化する。送信者は、次にその暗号化されたメッセージと MAC を受信者に送信する。受信者は、受信されたメッセージと MAC を復号し、MAC アルゴリズムと送信者の MAC 鍵を用いて MAC の自分のバージョンを計算する。受信者は、計算した MAC が送信者の送信した MAC と一致するかを検証する。

二つのタイプの構築が TLS の MAC アルゴリズムで使用される。TLS のすべてのバージョンはネゴシエートされた暗号スイートによって規定されるハッシュアルゴリズムを用いた鍵付ハッシュメッセージ認証(HMAC)の使用をサポートする。HMAC を用いて、サーバからクライアントへのメッセージの MAC がサーバ書込み MAC 鍵によって鍵付とされ、クライアントからサーバへのメッセージの MAC がクライアント書込み MAC 鍵で鍵付とされる。これらの MAC 鍵は共有されるマスタシークレットから導出される。

TLS 1.2 は CBC-MAC 付きのカウンター(CCM)およびガロアカウンターモード(GCM)のような、AEAD 暗号モードのサポートを、完全性と機密性を提供する別の方法として、追加した。AEAD モードにおいて、送信者は自身の書込み鍵を暗号化と完全性保護の両方に使用する。クライアントとサーバの書込み MAC 鍵は使用されない。受信者はメッセージを復号し、完全性情報を検証する。送信者と受信者の両方がこれらの操作を実行するために送信者の書込み鍵を使用する。

2.5 認証

サーバ認証は、サーバがハンドシェイク中に提示する、サーバの公開鍵証明書を用いてクライアントによって実行される。サーバ認証の暗号操作の正確な性質はネゴシエートされた暗号スイートおよび拡張に依存する。ほとんどの場合(例．鍵配送のための RSA、DH および ECDH)、認証は証明書の中で提示されるディジタル署名の検証を通して明示的に行われ、マスタシークレットの確立中にクライアントによるサーバ公開鍵の使用によって暗黙的に行われる。Finished メッセージの成功は両方の当事者が同じマスタシークレットを計算した結果、サーバは鍵確立で使用された公開鍵に対応した既知のプライベート鍵を持たなければならない。

クライアント認証はオプションであり、サーバの要求時のみ発生する。クライアント認証はクライアントの公開鍵証明書に基づく。クライアント認証の暗号操作の正確な性質は、ネゴシエートされた暗号スイートの鍵交換アルゴリズムとネゴシエートされた拡張に依存する。例えばクライアントの公開鍵証明

書が RSA 公開鍵を含むとき、クライアントはハンドシェイクメッセージの一部にその公開鍵に対応するプライベート鍵で署名し、サーバはクライアントを認証するためその公開鍵で署名を検証する。

2.6 耐リプレイ

メッセージの完全性保護されたエンベロープは、単純増加するシーケンス番号を含む。一度メッセージ完全性が検証されると、現在のメッセージのシーケンス番号は以前のメッセージのシーケンス番号と比較される。現在のメッセージのシーケンス番号は、メッセージをさらに処理するために、以前のメッセージのシーケンス番号よりも大きくなければならない。

2.7 鍵管理

サーバ公開鍵証明書と対応するプライベート鍵、およびオプションでクライアントの公開鍵証明書と対応するプライベート鍵は、選択された暗号スイートによって指示される鍵交換アルゴリズムに従って、プリマスタシークレットの確立において使用される。プリマスタシークレット、サーバランダム、およびクライアントランダムは、マスタシークレットを決定するために使用され、次にセッション対称鍵を導出するために使用される。

サーバのプライベート鍵のセキュリティは TLS のセキュリティにとって重要である。サーバのプライベート鍵が弱いか、第三者によって取得されることが可能な場合、第三者はすべてのクライアントに対してサーバとしてなりすましすることが可能となる。同様に、クライアントによって信頼される認証局(CA)から正当なサーバ名で第三者が自身のプライベート鍵に対応する公開鍵についての公開鍵証明書を取得可能な場合、第三者はクライアントに対してサーバとしてなりすましが可能である。これらの懸念を軽減するための要求事項と推奨事項はこれらのガイドラインにおいて後に記述される。

クライアントについても同様な脅威が存在する。クライアントのプライベート鍵が弱い場合、または第三者によって取得される可能性がある場合、第三者は、サーバに対してクライアントに成りすまし可能である。同様に、第三者が自身のプライベート鍵に対応する公開鍵のための公開鍵証明書をクライアントの名前でサーバによって信頼される CA から取得可能な場合、第三者はサーバに対してクライアントとしてなりすまし可能である。これらの懸念を軽減するための要求事項と推奨事項はこれらのガイドラインにおいて後に記述される。

クライアントおよびサーバによって生成される乱数はセッション鍵のランダムさに寄与するので、クライアントとサーバはそれぞれ少なくとも 112 ビットセキュリティ[6]を持つ疑似乱数を生成する能力を持たなければならない。これらのランダムな値から導出されたさまざまな TLS セッション鍵とその他のデータは、セッションの間中、有効である。なぜなら、セッション鍵は、アクティブな TLS セッション中に交換されるメッセージを保護するためのみに使用され、任意の保存データを保護するために使用されず、TLS セッション鍵を回復するような要求事項がないからである。しかし、サーバとクライアントは、セッション再開時の無視できないオーバーヘッドを軽減するために、マスタシークレットをキャッシュしてもよいし(また、しばしばキャッシュする)。クライアントとサーバの両方が以前のセッションからのマスタシークレットと関連するセッション ID をキャッシュに持っている場合、省略されたハンドシェイクがセッションを再開するために使用されることが可能である。再開されたセッションは以前のセッションと同じネゴシエートされたパラメータを使用するが、マスタシークレットと新しいサーバランダム値とクライアントランダム値から導出された新しいセッション鍵を使用する。何らかの合理的なタイムアウト期間の後、マスタシークレットはサーバとクライアントの両方において破棄されるべきである。セッション鍵を含めたすべての状態変数は、セッションが終了するときに破棄される。プロトコル実装は、ランダム値、プリマスタシークレットおよびセッション鍵等の、鍵材料の再利用がないこ

---

6 Bits of security (セキュリティ強度のビット数)は SP800-57 Part1 [SP800-57p1]、セクション 5.6 で記述された承認されたアルゴリズムによって提供された。

とを保証するため、オペレーティングシステムに信頼を置く。

3　TLS サーバの最小限の要求事項

本セクションは、これらのガイドラインを満たすためにサーバが実装しなければならないような最小限の要求事項を提供する。要求事項は、以下のセクションにおいて、体系付けられている：TLS プロトコルバージョンサポート；サーバ鍵と証明書；暗号サポート；TLS 拡張サポート；クライアント認証；セッション再開；圧縮方法；および運用上の検討事項。具体的な要求事項は、実装上の要求事項または設定上の要求事項のいずれかとして記述される。実装要求事項は、連邦政府機関が、必要な機能を含む場合を除き、TLS サーバの実装を調達**してはならない**こと、または要求事項を満たすために商用製品を追加できることを示す。設定要求事項は、TLS サーバ管理者が特定の機能が有効化されていること、または何らかの場合に、もしあれば、適切に設定されることを、検証することが要求されることを示す。

3.1　プロトコルバージョンサポート

TLS バージョン 1.1 は、最低限、TLS プロトコルのバージョン 1.0 に対するさまざまな攻撃を軽減するために、要求される。TLS バージョン 1.2 のサポートは強く推奨される。

政府専用アプリケーションをサポートするサーバは、TLS 1.1 をサポートするよう設定されなければならない、かつ TLS 1.2 をサポートするよう設定される**べきである**。これらのサーバは、TLS 1.0、SSL 2.0、または SSL 3.0 をサポート**してはならない**。TLS バージョン 1.1 および 1.2 は、メジャー番号およびマイナー番号(3,2)と(3,3)でそれぞれ[7]表される。政府機関は 2015 年 1 月 1 日までに TLS1.2 をサポートするような移行計画を策定**しなければならない**。

市民または商用関連のアプリケーションをサポートするサーバは、バージョン 1.1 をサポートするよう設定され**なければならない**、またバージョン 1.2 をサポートするよう設定できる**べきである**。これらのサーバは、市民および企業とのやりとりを可能とするため、TLS バージョン 1.0 をサポートするよう設定してもよい。これらのサーバは、SSL 3.0 またはそれ以前のものをサポート**してはならない**。TLS 1.0 がサポートされる場合、TLS 1.1 および 1.2 の使用が TLS 1.0 よりも優先**されなければならない**。

いくつかのサーバ実装が正しくないバージョンのネゴシエーションを実装していることが知られている。例えば、クライアントが TLS 1.0 よりも新しいバージョンを提案するとき、コネクションを終了するような TLS 1.0 サーバがある。TLS バージョンのネゴシエーションを誤って実装したサーバは使用**されてはならない**。

3.2　サーバ鍵と証明書

TLS サーバは、一つ以上の公開鍵証明書と対応するプライベート鍵と共に設定**されなければならない**。TLS サーバ実装は、アルゴリズムと鍵長の俊敏性をサポートするため、複数のサーバ証明書を対応するプライベート鍵と共にサポートする**べきである**。

承認された暗号のための要求事項を満たすことが可能な TLS サーバ証明書には 6 つのオプションがある：RSA 鍵暗号化証明書；RSA 署名証明書；楕円曲線ディジタル署名アルゴリズム(ECDSA)署名証明書；ディジタル署名アルゴリズム(DSA)[8]署名証明書；Diffie-Hellman 証明書；および ECDH 証明書。

最小限、この仕様に適合する TLS サーバは、RSA 鍵暗号化証明書と共に設定**されなければならない**、また ECDSA 署名証明書または RSA 署名証明書と共に設定される**べきである**。サーバが RSA 署名証明書と共に設定されない場合、ECDSA 証明書における署名および公開鍵のためのスイート B 指定の曲線を用いた ECDSA 署名証明書が使用される**べきである**[9]。

---

[7] 歴史的に TLS 1.0 は、SSL 3.1 と揃えるため、メジャー、マイナーの組 (3,1) と割り付けられた。

[8] TLS 暗号スイートの名前において、DSA は歴史的な理由により、DSS (Digital Signature Standard) として参照される。

[9] スイート B 曲線は、P-256 および P-384 として知られている。これらの曲線は[FIPS186-4]で定義されており、スイー

TLS サーバは、自己署名ではなく、CA によって発行された証明書と共に設定**されなければならない**。さらに、TLS サーバ証明書は、証明書失効リスト(CRL)[RFC5280]またはオンライン証明書状態プロトコル(OCSP)[RFC6960]応答のいずれかにおける失効情報を公表するような CA によって発行**されなければならない**。失効情報の情報源は、相互運用性を促進するため、CA 発行の証明書の適切な拡張に含**まれなければならない**。

複数の CA によって複数の証明書が発行された TLS サーバは、セクション 3.4.1.4 で記述されるとおり、クライアントが規定した「TrustedCA Keys」拡張に基づいて、適切な証明書をひとつ選択することが可能である。複数の名前の複数の証明書が発行された TLS サーバは、セクション 3.4.1.3 で記述されるとおり、クライアントが規定した「Server Name」拡張に基づいて、適切な証明書を選択可能である。TLS サーバは、同じ名前形式の複数のサーバ名(例．DNS Name)または複数の名前形式の複数のサーバ名(例．DNS 名、IP アドレス、等)をサポートするため、サーバ証明書の Subject Alternative Name 拡張において複数の名前についても含んでもよい。

セクション 3.2.1 は、サーバ証明書の詳細なプロファイルを規定する。DSA、DH および ECDH 証明書の基本的なガイドラインが提供される；これらのアルゴリズムが将来幅広く使用される場合さらに詳しいプロファイルが提供されるかもしれない。セクション 3.2.2 は、失効チェックについての要求事項を規定する。システム管理者は、証明書のための適切な情報源を識別するためにこれらのセクションを使用**しなければならない**。セクション 3.5.4 は、「ヒントリスト」の要求事項を規定する。

3.2.1 サーバ証明書プロファイル

このセクションで記述されるサーバ証明書プロファイルは、サーバ証明書のフォーマットについての要求事項と推奨事項を提供する。これらのガイドラインについて、TLS サーバ証明書は、X.509 バージョン 3 証明書**でなければならない**；証明書に含まれる公開鍵と署名は少なくとも 112 bits のセキュリティを持**たなければならない**。証明書は、公開鍵[10]と一貫するアルゴリズムを用いて署名**されなければならない**。

- RSA(鍵暗号化または署名)、ECDSA、または DSA 公開鍵を含む証明書は、それぞれ同じ署名アルゴリズム、を用いて署名**されなければならない**；
- Diffie-Hellman 公開鍵を含む証明書は、DSA を用いて署名**されなければならない**；そして
- ECDH 公開鍵を含む証明書は、ECDSA を用いて署名**されなければならない**。

拡張された鍵用途拡張は、証明書の鍵が使用される操作を制限する。サーバ認証用に特別に拡張された鍵用途拡張があり、サーバは、それをサポートするように設定される**べきである**。拡張された鍵用途拡張の使用は、何らかのクライアントが拡張された鍵用途拡張の存在を要求するかもしれないので、サーバ認証の成功を促進するだろう。拡張された鍵用途拡張は、証明書が、コード署名のような、その他の目的で使用されるよう意図されていないことを示す。Subject Alternative Name フィールドでのサーバ DNS 名の使用は、証明書パス上の任意の名前制限が適切に実施されていることを保証する。

サーバ証明書プロファイルが表 3-1 に列挙されている。政府機関指定の証明書プロファイル要求事項の欠如において、この証明書プロファイルはサーバ証明書のために使用される**べきである**。

ECDH については、アルゴリズム object identifier(OID)と署名 OID が ECDSA のそれらと同一であることに注意すること。相互運用性の理由で、アルゴリズム OID は変更されず、鍵用途拡張は公開鍵が鍵共有

ト B にそれらを含めることは[RFC6460]で記述されている。

10 アルゴリズム依存の規則は公開鍵とプライベート鍵ペアの生成について存在する。DH および ECDH 鍵ペアの生成に関するガイダンスについては、[SP800-56A]を参照。RSA 鍵エアの生成に関するガイダンスについては、[SP800-56B]を参照。DSA および ECDSA 鍵ペアの生成に関するガイダンスについては[FIPS186-4]を参照。

または署名検証のために使用されるかどうかを決定する。

表 3-1： TLS サーバ証明書プロファイル

Field フィールド	Critical	Value 値	Description 説明
Version バージョン	N/A	2	Version 3 バージョン 3
Serial Number シリアル番号	N/A	Unique positive integer ユニークな正の整数	Must be unique ユニークでなければならない
Issuer Signature Algorithm 発行者署名アルゴリズム	N/A	*Values by certificate type:証明書タイプによる値*	
		sha256WithRSAEncryption {1 2 840 113549 1 1 11}, or stronger	RSA key encipherment certificate, RSA signature certificate RSA 鍵暗号化証明書、RSA 署名証明書
		ecdsa-with-SHA256 {1 2 840 10045 4 3 2}, or stronger	ECDSA signature certificate, ECDH certificate ECDSA 署名証明書、ECDH 証明書
		id-dsa-with-sha256 {2 16 840 1 101 3 4 3 2}, or stronger	DSA signature certificate, DH certificate DSA 署名証明書、DH 証明書
Issuer Distinguished Name 発行者識別名	N/A	Unique X.500 Issuing CA DN ユニークな X.500 発行 CA 識別名	Single value shall be encoded in each Relative Distinguished Name (RDN). All attributes that are of directoryString type shall be encoded as a printable string. 一つの値がそれぞれの関連識別名(RDN)においてエンコードされなければならない。directoryString タイプであるすべての属性が PrintableString としてエンコードされなければならない。
Validity Period 有効期間	N/A	3 years or less 3 年以下	Dates through 2049 expressed in UTCTime UTCTime で表現された 2049 年までの日付
Subject Distinguished Name サブジェクト識別名	N/A	Unique X.500 subject DN per agency requirements 政府機関要求事項ごとのユニークな X.500 サブジェクト識別名	Dates value shall be encoded in each RDN. All attributes that are of directoryString type shall be encoded as a printable string. CN={Host IP Address | Host DNS Name}
Subject Public Key Information サブジェクト公開鍵情報	N/A	*Values by certificate type:証明書タイプによる値*	
		rsaEncryption (1 2 840 113549 1 1 1}	RSA key encipherment certificate, RSA signature certificate 2048-bit RSA key modulus, or other approved lengths as defined in [SP800-56B] and [SP800-57p1]] Parameters: NULL. RSA 鍵暗号化証明書、RSA 署名証明書 2048-bitRSA 鍵モジュラス、または[SP800-56B]および[SP800-57p1]で定義されたとおりのその他の承認された長さ パラメータ：なし
		ecPublicKey {1 2 840 10045 2 1}	ECDSA signature certificate, or ECDH certificate Parameters: namedCurve OID for names curve specified in FIPS 186-4. The curve shall be P-256 or P-384 SubjectPublicKey: Uncompressed EC Point. ECDSA 署名証明書、または ECDH 証明書

Field フィールド	Critical	Value 値	Description 説明
			パラメータ：FIPS 186-4 で規定された曲線目の namedCurve OID。曲線は P-256 または P-384 でなければならない。SubjectPublicKey:非圧縮の EC Point。
		id-dsa {1 2 840 10040 4 1}	DSA signature certificate Parameters: p, g, q (2048 bit large prime, i.e., p) DSA 署名証明書 パラメータ： p, g, q (2048bit large prime, 即ち、p)
		dhpublicnumber {1 2 840 10046 2 1}	DH certificate Parameters: p, g, q (2048 bit large prime, i.e., p) DH 証明書 パラメータ：p, g, q (2048 bit large prime, 即ち、p)
Issuer's Signature 発行者の署名	N/A	*Values by certificate type:証明書タイプによる値*	
		sha256WithRSAEncription {1 2 840 113549 1 11}, or stronger	RSA key encipherment certificate, RSA signature certificate RSA 鍵暗号化証明書、RSA 署名証明書
		ecdsa-with-SHA256 {1 2 840 10045 4 3 2}, or stronger	ECDSA signature certificate, ECDH certificate ECDSA 署名証明書、ECDH 証明書
		id-dsa-with-sha256 {2 16 840 1 101 3 4 3 2}, or stronger	DSA signature certificate, DH certificate DSA 署名証明書、DH 証明書
Extensions			
Authority Key Identifier オーソリティ鍵識別子	No	Octet string	Same as subject key identifier in Issuing CA certificate Prohibited: Issuer DN, Serial Number tuple 発行 CA 証明書におけるサブジェクト鍵識別子と同じ 禁止：発行者識別名、シリアル番号タプル(組)
Subject Key Identifier サブジェクト鍵識別子	No	Octet String	Same as in PKCS-10 request or calculated by the Issuing CA PKCS-10 リクエスト内と同じまたは発行 CA により計算されたものと同じ
Key Usage 鍵用途	Yes	*Values by certificate type:証明書タイプによる値*	
		key Encipherment	RSA key encipherment certificate RSA 鍵暗号化証明書
		digitalSignature	RSA signature certificate, ECDSA signature certificate, or DSA signature certificate RSA 署名証明書、ECDSA 署名証明書、または DSA 署名証明書
		keyAgreement	ECDH certificate, DH certificate ECDH 証明書、DH 証明書
Extended key Usage 拡張鍵用途	No	id-kp-serverAuth {1 3 6 1 5 5 7 3 1}	Required 必須
		id-kp-clientAuth {1 3 6 1 5 5 7 3 2}	Optional オプション

Field フィールド	Critical	Value 値	Description 説明
			Prohibited: anyExtendedKeyUsage, all others unless consistent with key usage extension 禁止：anyExtendedKeyUsage、鍵用途拡張と一貫しないその他すべて
Certificate Policies 証明書ポリシー	No	Per agency X.509 certificate policy	
Subject Alternative Name サブジェクト別名	No	DNS Host Name or IP Address if there is no DNS name assigned	Multiple SANs are permitted, e.g., for load balanced environments. 複数の SAN が許容される、例. 負過バランス環境
Authority Information Access オーソリティ情報アクセス	No	id-ad-caIssuers	Required. Access method entry contains HTTP URL for certificates issued to Issuing CA 必須。アクセス方法入力には発行 CA へ発行された証明書への HTTP URL が含まれる
		id-ad-ocsp	Optional. Access method entry contains HTTP URL for the Issuing CA OCSP Responder オプション。アクセス方法入力には発行 CA の OCSP レスポンダの HTTP URL が含まれる
CRL Distribution Points CRL 配付ポイント	No	See comments	Optional. HTTP value in distributionPoint field pointing to a full and complete CRL. Prohibited: reasons and cRLIssuer fields, and nameRelativetoCRLIssuer CHOICE オプション。すべての CRL および完全な CRL を指す distributionPoint フィールドにおける HTTP 値 禁止：reasons and cRLIssuer fields、nameRelativetoCRLIssuerCHOICE

3.2.2 クライアント証明書の失効状態情報の取得

サーバは、クライアント認証が使用されるとき、クライアント証明書の失効チェックを実行**しなければならない**。失効情報は、以下のロケーションの一つまたは複数からサーバによって取得**されなければならない**：

1. サーバのローカルストアにおける、証明書失効リスト(CRL) または OCSP [RFC 6960] レスポンス；
2. ローカル設定された OCSP レスポンダからの OCSP レスポンス；
3. クライアント証明書のオーソリティ情報アクセス拡張の OCSP フィールドで識別される OCSP レスポンダロケーションからの OCSP レスポンス；または
4. クライアント証明書の CRL 配付ポイント拡張からの CRL

ローカルストアが現在の、または説得力のある [11]CRL または OCSP レスポンスを持っていないとき、また OCSP レスポンダおよび CRL 配付ポイントが TLS セッション確立時に利用できないまたはアクセス

11 CRL は、「CRL 適用範囲」が問合せ中の証明書に対して適切であるとき、「cogent (適切)」であるとみなされる。「CRL 適用範囲は、[RFC5280]で定義される。

できないとき、サーバは、コネクションを拒否または失効されたかまたは危殆化した可能性のある証明書を受けいれるか、のいずれかをしなければならない。この状況で証明書を受け入れるか、拒否するかの決定は、政府機関のポリシーに従ってなされるべきである。

3.2.3 サーバ公開鍵証明書の保証

サーバ公開鍵証明書がクライアントによって検証された後、サーバ公開鍵証明書を発行するために使用されるようなポリシー、手順およびセキュリティ管理策に基づき信頼されるかもしれない。サーバは、X.509バージョン3公開鍵証明書を所持することを要求される。ポリシー、手順および管理策が、[RFC5280]で規定され、[RFC6818]で更新された、certificatePolicies拡張を用いる証明書にオプションで書かれる。使用されるとき、一つまたは複数の証明書ポリシーOIDがこの拡張で行使される。それぞれの証明書ポリシーOIDに関連する実際のポリシーおよび手順およびセキュリティ管理策は、証明書ポリシーに書かれる。政府機関特有のポリシーが無い場合は、連邦政府機関は、共通ポリシー[COMMON]を使用**しなければならない**。

PKIのセキュアな運用を念頭に置いて設計されたような証明書ポリシーの使用と規定された証明書ポリシーの遵守は、発行CAが危殆化する可能性がある、または登録システム、要員またはプロセスが正当なエントリーの名前において許可されない証明書を取得して危殆化し、その結果としてクライアントを危殆化するような脅威 (訳注：脅威の顕在化) を軽減する。これを念頭に置いて、CAブラウザフォーラムという民間組織が、この分野でいくつかの取組みを行った。拡張検証ガイドライン[EVGUIDE]として、ガイドラインが最初に発行された。別の取組みでは、CAブラウザフォーラムは、それらのCAとトラストアンカがブラウザトラストアンカに留まるための公的に信頼されるCAから証明書を発行するための要求事項を発行した[CABBASE]。

[RFC5280]によって義務付けられるとおりにX.509証明書ポリシー処理を実行しないようなTLSクライアントがあることに注意するべきである。したがって、それらは、そのポリシーで規定される保証レベルに基づくTLSサーバ証明書を受け入れたり拒否したりできない。これは、詐称した証明書の受け入れという結果となったり、意図されていないものへの利用者データを暴露したりするかもしれない。連邦政府およびCAブラウザフォーラムは、[COMMON]、[EVGUIDE]および[CAABASE]のセキュリティ要件がすべてのCAにそれらの範囲の下で適用され、ポリシー処理の欠如を軽減することを望んでいる。

CA または X.509 証明書登録システム、プロセスまたは要員の危殆化に関連するリスクをさらに軽減するために、いくつかの概念が開発中である。これらの新たな概念は附属書 D でさらに議論される。

3.3 暗号サポート

TLS の暗号サポートは、さまざまな暗号スイートの使用を通して提供される。暗号スイートは、鍵交換のため、およびアプリケーションデータへの機密性と完全性サービスを提供するためのアルゴリズムの集合を規定する。暗号スイートネゴシエーションは、TLS ハンドシェイクプロトコル中に発生する。クライアントは、サポーとする暗号スイートをサーバに提示する、またサーバはセッションデータをセキュアにするため、それらの一つを選択する。

暗号スイートは、以下の形式を持つ：

TLS_*KeyExchangeAlg*_WITH_*EncryptingAlg*_*MessageAuthenticationAlg*

例えば、暗号スイート TLS_RSA_WITH_AES_128_CBC_SHA は、鍵交換に RSA を使い、暗号化のために AES-128 を cipher block chaining (CBC) モードで使い、メッセージ認証が HMAC_SHA[12]を用いて実行される。暗号スイート実装についてさらなる情報については、附属書 B を参照。

[12] SHA は、SHA-1 ハッシュアルゴリズムの使用を示す。

3.3.1 暗号スイート

サーバは、完全に承認されたアルゴリズムからなる暗号スイートのみを使用するよう設定され**なければならない**。一般的用途の受け入れ可能な暗号スイートの完全なリストが本セクションで提供され、証明書タイプと TLS プロトコルバージョンによりグループ化されている。

閉鎖的な環境等、何らかの状況において、事前共有鍵の使用が適切かもしれない。事前共有鍵は TLS セッションの開始前にすでに環境が整っているような対称鍵であり、プリマスタシークレットの導出で使用される。事前共有鍵環境で受け入れ可能な暗号スイートについては、附属書 C を参照。

相互運用性を最大化するため、TLS サーバ実装は、以下の暗号スイートをサポートし**なければならない**:

- TLS_RSA_WITH_3DES_EDE_CBC_SHA[13]
- TLS_RSA_WITH_AES_128_CBC_SHA[14]

さらに、TLSサーバ実装は、以下の暗号スイートをサポートするべきである：

- TLS_RSA_WITH_AES_256_CBC_SHA
- TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA[15]
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
- TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA

マスタシークレットを確立するために、一時的な鍵が使用されるとき、それぞれの一時的鍵ペア(即ち、サーバの一時的鍵ペアとクライアントの一時的鍵ペア) は、少なくとも 112 bit のセキュリティを持た**なければならない**。

TLS バージョン 1.2 は認証付き暗号化モードのサポートと、TLS の以前のバージョンではサポートされていないような、SHA-256 と SHA384 ハッシュアルゴリズムのサポートを追加している。これらの暗号スイートは、[RFC5288]と[RFC5289]で記述されている。上記暗号スイートのサポートに加えて、TLS 1.2 は、以下の暗号スイートをサポートするよう設定されなければならない：

- TLS_RSA_WITH_AES_128_GCM_SHA256

TLS 1.2 サーバは、以下の暗号スイートをサポートするように設定されるべきである：

- TLS_RSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
- TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
- TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

---

13 この暗号スイートのサポートは、TLS 1.1 [RFC4346]で必須である。

14 この暗号スイートのサポートは、TLS 1.2 [RFC5246]で必須である。

15 TLS バージョン 1.0 と 1.1 では、DHE と ECDHE 暗号スイートは ServerKeyExchange メッセージにおける一時的なパラメータ(鍵を含めて) 上で署名生成に SHA-1 を使用する。[SP800-131A]は、ディジタル署名生成の SHA-1 の使用は 2013 年以降許容されないことが述べられている。一時的鍵のランダムな性質により、サードパーティは有効な衝突を起こすことができない。クライアントランダムとサーバランダムによって、クライアント、またはサードパーティは、将来の衝突でクライアントまたはサーバとしてなりすましするために衝突するメッセージのセットを使用できない。ハンドシェイク中のサードパーティによるパラメータへの任意の改ざんは、最終的にコネクション櫛比の結果となるだろう。これらの理由により、SHA-1 は TLS における一時的なパラメータ上のディジタル署名生成用に許容される。

NIST は、後日、追加の必須または推奨される暗号スイートを定義するかもしれない。

サーバは、少なくとも 112 bits のセキュリティを提供するような署名を含む有効な証明書を持つように暗号スイートのみをサポートするように設定され**なければならない**。以下の暗号スイート表は、証明書タイプと TLS プロトコルバージョンによってグループ化されている。これらの表の暗号スイートは、サポートされ**なければならない**、および**されるべきでる**、およびサポートされてもよい暗号スイートを含んでいる。承認されたアルゴリズムからなる暗号スイートのみが受け入れ可能であり、本セクションに列挙されている。本セクションまたは附属書 C に現れないような暗号スイートは、使用**されてはならない**。

推奨される暗号スイートを列挙している以下の表において、**太字**で表される暗号スイートは、サポートされ**なければならない**、*イタリックフォント*で表される暗号スイートはサポートされる**べきである**、また標準フォントで表される暗号スイートはサポートされてもよい。

表 3-2 は、RSA プライベート鍵と対応する RSA 証明書を用いて設定された TLS サーバの受け入れ可能な暗号スイートの三つのカテゴリー(**しなければならない**、*するべきである*、してもよい)を識別する。表 3-3 は、TLS バージョン 1.2 サーバの追加の RSA 暗号スイートを三つのカテゴリーについて識別する。RSA 証明書を持つようなサーバは、表 3-2 または表 3-3 に表れる任意の暗号スイートをサポートしてもよい。RSA 証明書における鍵用途拡張は、鍵交換を行うために RSA 鍵配送を使用するような暗号スイートの鍵暗号化を規定しなければならない、また鍵用途拡張は鍵交換のために ECDHE を用いる暗号スイートのディジタル署名を規定**しなければならない**。

表 3-2：RSA サーバ証明書の暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF[16]
TLS_RSA_WITH_3DES_EDE_CBC_SHA	**RSA**	**3DES_EDE_CBC**	**SHA-1**	**Per RFC**
TLS_RSA_WITH_AES_128_CBC_SHA	**RSA**	**AES_128_CBC**	**SHA-1**	**Per RFC**
TLS_RSA_WITH_AES_256_CBC_SHA	*RSA*	*AES_256_CBC*	*SHA-1*	*Per RFC*
TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA	*ECDHE*	*3DES_EDE_CBC*	*SHA-1*	*Per RFC*
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA	*ECDHE*	*AES_128_CBC*	*SHA-1*	*Per RFC*
TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA	ECDHE	AES_256_CBC	SHA-1	Per RFC

表 3-3：RSA サーバ証明書の追加の TLS 1.2 暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_RSA_WITH_AES_128_GCM_SHA128	**RSA**	**AES_128_GCM**	**N/A**	**SHA-256**
TLS_RSA_WITH_AES_256_GCM_SHA384	*RSA*	*AES_256_GCM*	*N/A*	*SHA-384*
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256	*ECDHE*	*AES_128_CBC*	*N/A*	*SHA-256*
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256	*ECDHE*	*AES_128_GCM*	*N/A*	*SHA-256*
TLS_RSA_WITH_AES_128_CBC_SHA256	RSA	AES_128_CBC	SHA-256	SHA-256
TLS_RSA_WITH_AES_256_CBC_SHA256	RSA	AES_256_ CBC	SHA-256	SHA-256
TLS_RSA_WITH_AES_128_CCM[17]	RSA	AES_128_CCM	N/A	SHA-256
TLS_RSA_WITH_AES_256_CCM	RSA	AES_256_CCM	N/A	SHA-256

[16] TLS バージョン 1.0 と 1.1 において、PRF で使用されるハッシュ関数は、[RFC2246]と[RFC4346]で定義されるとおりの MD5 と SHA-1 の並行適用となる。TLS 1.2 については、PRF ハッシュ関数は、特に指定のない限り、SHA-256 となる。

[17] AES_CCM 暗号スイートは、[RFC6655]で定義される。

表 3-4 は、楕円曲線プライベート鍵と対応する ECDSA 証明書と共に設定されるような TLS サーバの暗号スイートとの二つのカテゴリ(*するべきである*、してもよい)を識別する。これらの暗号スイートは、[RFC4492]で記述されている。表 3-5 は、TLS バージョン 1.2 サーバの[RFC5289]で記述されるような、ECDSA 暗号スイートの追加の二つのカテゴリ(*するべきである*、してもよい)を識別している。ECDSA 証明書と共に設定されるサーバは、表 3-4 または表 3-5 で列挙された任意の暗号スイートをサポートすることができる。

表 3-4：ECDSA サーバ証明書の暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA	*ECDHE*	*3DES_EDE_CBC*	*SHA-1*	*Per RFC*
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA	*ECDHE*	*AES_128_CBC*	*SHA-1*	*Per RFC*
TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA	ECDHE	AES_256_CBC	SHA-1	Per RFC

表 3-5：ECDSA サーバ証明書の追加の TLS 1.2 暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256	*ECDHE*	*AES_128_CBC*	*SHA-256*	*SHA-256*
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256	*ECDHE*	*AES_128_GCM*	*N/A*	*SHA-256*
TLS_ECDHE_ECDSA_WITH_ AES_256_GCM_SHA384	*ECDHE*	*AES_256_GCM*	*N/A*	*SHA-384*
TLS_ECDHE_ECDSA_WITH_ AES_256_CBC_SHA384	ECDHE	AES_256_CBC	SHA-384	SHA-384

表 3-6 は、DSA プライベート鍵と対応する DSA 証明書と共に設定されるサーバによってサポートされてもよい暗号スイートを識別する。表 3-7 は、TLS バージョン 1.2 サーバによってサポートされもよい追加の DSA 暗号スイートを識別する。DSA 証明書と共に設定されるようなサーバは、表 3-6 と表 3-7 に列挙される任意の暗号スイートをサポートすることができる。

表 3-6：DSA サーバ証明書の暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA	DHE	3DES_EDE_CBC	SHA-1	Per RFC
TLS_DHE_DSS_WITH_AES_128_CBC_SHA	DHE	AES_128_CBC	SHA-1	Per RFC
TLS_DHE_DSS_WITH_ AES_256_CBC_SHA	DHE	AES_256_CBC	SHA-1	Per RFC

表 3-7：DSA サーバ証明書の追加の TLS1.2 暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_DHE_DSS_WITH_AES_128_CBC_SHA256	DHE	AES_128_CBC	SHA-256	SHA-256
TLS_DHE_DSS_WITH_ AES_256_CBC_SHA256	DHE	AES_256_CBC	SHA-256	SHA-256
TLS_DHE_DSS_WITH_AES_128_GCM_SHA256	DHE	AES_128_GCM	N/A	SHA-256
TLS_DHE_DSS_WITH_ AES_256_GCM_SHA384	DHE	AES_256_GCM	N/A	SHA-384

表 3-8 は、DH プライベート鍵と対応する DSA を用いて署名された DH 証明書と共に設定される TLS サーバによってサポートされてもよい暗号スイートを識別する。表 3-9 は、TLS 1.2 サーバ[RFC5246]、[RFC5288]によってサポートされるかもしれない追加の DH 暗号スイートを識別する。

表 3-8：DH サーバ証明書の暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_DH_DSS_WITH_3DES_EDE_CBC_SHA	DH	3DES_EDE_CBC	SHA-1	Per RFC
TLS_DH_DSS_WITH_AES_128_CBC_SHA	DH	AES_128_CBC	SHA-1	Per RFC
TLS_DH_DSS_WITH_ AES_256_CBC_SHA	DH	AES_256_CBC	SHA-1	Per RFC

表 3-9：DH サーバ証明書の追加の TLS 1.2 暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_DH_DSS_WITH_AES_128_CBC_SHA256	DH	AES_128_CBC	SHA-256	SHA-256
TLS_DH_DSS_WITH_ AES_256_CBC_SHA256	DH	AES_256_CBC	SHA-256	SHA-256
TLS_DH_DSS_WITH_AES_128_GCM_SHA256	DH	AES_128_GCM	N/A	SHA-256
TLS_DH_DSS_WITH_ AES_256_GCM_SHA384	DH	AES_256_GCM	N/A	SHA-384

表 3-10 は、楕円曲線プライベート鍵と対応する ECDSA を用いて署名された ECDH 証明書と共に設定されたサーバによってサポートされてもよい暗号スイートを識別する。表 3-11 は、TLS 1.2 サーバによってサポートされるかもしれない追加の ECDH 暗号スイートを識別する。これらの暗号スイートは[RFC5289]で定義される。

表 3-10：ECDH サーバ証明書の暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_ECDH_ECDSA_WITH_3DES_EDE_CBC_SHA	ECDH	3DES_EDE_CBC	SHA-1	Per RFC
TLS_ ECDH_ECDSA _WITH_AES_128_CBC_SHA	ECDH	AES_128_CBC	SHA-1	Per RFC
TLS_ ECDH_ECDSA _WITH_ AES_256_CBC_SHA	ECDH	AES_256_CBC	SHA-1	Per RFC

表 3-11：ECDH サーバ証明書の追加の TLS 1.2 暗号スイート

Cipher Suite Name	Key Exchange	Encryption	Hash Function for HMAC	Hash Function for PRF
TLS_ ECDH_ECDSA _WITH_AES_128_CBC_SHA256	ECDH	AES_128_CBC	SHA-256	SHA-256
TLS_ ECDH_ECDSA _WITH_ AES_256_CBC_SHA384	ECDH	AES_256_CBC	SHA-384	SHA-384
TLS_ ECDH_ECDSA _WITH_AES_128_GCM_SHA256	ECDH	AES_128_GCM	N/A	SHA-256
TLS_ ECDH_ECDSA _WITH_ AES_256_GCM_SHA384	ECDH	AES_256_GCM	N/A	SHA-384

附属書 B は、暗号スイート名前解釈についてのさらなる詳細を説明する。説明では暗号スイート名が使用されるが、実際のプロトコルは暗号スイートを識別するために割り付けられた番号を使用する。

暗号スイートをネゴシエートするとき、クライアントは、受け入れる暗号スイートのリストと共にハンドシェイクメッセージを送信する。サーバは、そのリストから選択を行い、受け入れる暗号スイートを示すようなハンドシェイクメッセージを送り返す。クライアントが最初に列挙した最も強度のある暗号スイートを持つリストを指定することができるが、サーバはクライアントによって提案された任意の暗号スイートを選択してもよい。ゆえに、ネゴシエーションは通常最も強い暗号スイートを決めるという保証はない。通常、一切の暗号スイートが無い場合、コネクションは中止される。

一時的な鍵を持つ DH と一時的な鍵を持つ ECDH(即ち､2 番目のニーモニックで DHE または ECDHE を持つもの)を用いる暗号スイートは、セッションの長期間機密性を保証するような完全前方秘匿性 [18]を

[18] 完全前方秘匿性は、セッション鍵の危殆化を引き起こさないような導出に続くセッション鍵の導出で使用される

提供する。これらの暗号スイートのサポートはこれらのガイドラインによって要求されないが、強く推奨される。

サーバまたはクライアント証明書、または証明書経路にあるような CA の最小鍵長を規定するようなメカニズムは一切ない。

3.3.1.1 実装上の検討事項

システム管理者は、暗号スイートの選択およびそれらの暗号スイートのみをサポートするようなアプリケーションを設定することの影響を十分に理解する必要がある。暗号のセキュリティ保証は、設定によってサポートされる最も弱い暗号スイートに限定される。実装を設定するとき、サポートされる暗号スイート選択に影響するようないくつかの要素がある。

[RFC4346]は、CBC 暗号スイート上のタイミング攻撃と、軽減する技術と共に、記述している。TLS 実装は、パディングエラーを示すための bad_record_mac エラーを使用し**なければならない**。実装は、パディングエラーが存在するかどうかにかかわらず MAC を計算し**なければならない**。

[RFC4346]で述べられた CBC 攻撃に追加して、Lucky 13 攻撃 [Lucky13]は、一定時間復号ルーチンがタイミング攻撃を防止するためにも必要であることを実証している。TLS 実装は、一定時間復号、または、ほぼ一定時間復号をサポートする**べきである**。

CBC ベースの攻撃は、TLS 1.2 でサポートされるような、AEAD 暗号スイート(例．GCM、CCM)を用いることによって防止されることが可能であることに注意すること。

3.3.1.1.1 アルゴリズムサポート

多くの TLS サーバとクライアントは、RC4 [Schneier96] 暗号スイートをサポートしている。RC4 は、承認されたアルゴリズムではない。サーバが RC4 暗号スイートをサポートするように設定された場合、それらは承認されたアルゴリズムからなる推奨される暗号スイートを越えて選択されるかもしれない。ゆえに、サーバが推奨される暗号スイートのみを使用するよう設定されることは重要である。

サーバ実装は、優先順序を規定することをサーバ管理者に許容しないかもしれない。このようなサーバにおいて、サーバが暗号化のために承認されたアルゴリズムを使用することを保証するための唯一の方法はその他のアルゴリズム(RC4 およびカメリア[RFC3713]等)を使用するような暗号スイートを無効化することである。

3.3.1.1.2 暗号スイートの範囲

暗号アルゴリズムの選択は、システム全体にわたるかもしれないし、またいくつかの実装に特有なアプリケーションではないかもしれない。例えば、システム上のあるアプリケーションのためのアルゴリズムを無効化することは、そのシステム上のすべてのアプリケーションのそのアルゴリズムを無効化するかもしれない。

3.3.2 認証された暗号

サーバで使用される暗号モジュールは、FIPS 140 認証済の暗号モジュールで**なければならない**。設定された暗号スイートに含まれるすべての暗号アルゴリズムは乱数生成器と同様に、認証の適用範囲内で**なければならない**。TLS 1.1 疑似乱数関数(PRF)が、一つのハッシュ関数が破られた場合にセキュリティが危殆化しないように、MD5 と SHA-1 を並行して使用することに注意すること。MD5 は承認されたアルゴリズムではないが、TLS 1.1 PRF は[FIPS140Impl]および[SP800-135]において受け入れ可能として規定している。TLS 1.1 において、SHA-1 の使用は、一時的鍵を署名する特定の場合およびクラ

長期間使用のプライベート鍵の危殆化の条件である。

イアント認証の署名のために受け入れられていることが判っていることに注意すること。これは、セクション 3.3.1 の脚注でさらに説明されているとおり、第三者が検出されない衝突を引き起こすことができないこと、クライアントサーバが引き起こされた衝突を悪用できないという事実により受け入れられている。TLS 1.2 において、PRF におけるデフォルトハッシュ関数は、SHA-256 である。上記の具体的な場合について列挙した SHA-1 の例外事項以外、使用されるすべての暗号は、少なくとも 112bits のセキュリティを提供し**なければならない**。すべてのサーバとクライアント証明書は、112bits のセキュリティを提供するような公開鍵を含ま**なければならない**。すべてのサーバとクライアント証明書と証明書経路にある証明書は、少なくとも 112bits のセキュリティを提供するような鍵ペアと SHA224 またはそれよりも強度のあるハッシュアルゴリズムを用いて署名され**なければならない**。クライアントとサーバで使用されるすべての一時的鍵は、少なくとも 112bits のセキュリティを提供し**なければならない**。TLS データを保護するために使用されるすべての対称アルゴリズムは、少なくとも 112bits のセキュリティを提供する様な鍵を使用し**なければならない**。

乱数生成器は、NIST 暗号アルゴリズム認証プログラム（CAVP）のもと、[SP800-90A]に従って試験され、検証され**なければならない**、また、この試験の成功した結果は暗号モジュールの FIPS140 検証証明書に示され**なければならない。**

ServerHello メッセージで送信されるサーバランダム値は、4-byte タイムスタンプ[19]値と 28-byte ランダム値を含む。認証された乱数生成器は、サーバランダム値の 28-byte ランダム値を生成するために使用され**なければならない**。認証された乱数生成器は、サーバランダム値の 4-byte タイムスタンプを訂正するために使用されるべきである。

3.4 TLS 拡張サポート

いくつかの TLS 拡張が[RFC6066]に記述されている。サーバは、セクション 3.4.3 で規定されるとおり抑制されるものを除き、これらの拡張をサポートするよう奨励される。追加の拡張が[RFC4492]、[RFC5246]、および[RFC5746]に記述されている。このセクションは、市販の利用可能な TLS サーバとクライアントにおいて普及しているものとして、連邦政府機関が使用し**なければならない**、する**べきである**、**するべきでない**ような TLS 拡張のサブセットについての推奨事項を含む。

あるサーバは、任意の TLS 拡張が ClientHello メッセージに含まれる場合にコネクションを拒否する。TLS 拡張を適切に取り扱わないようなサーバとの相互運用性は、クライアントによる複数のコネクション試行を要求するかもしれない。

3.4.1 必須の TLS 拡張

サーバは以下の TLS 拡張をサポートし**なければならない**。

1. 再ネゴシエーション指示
2. 証明書状態要求
3. サーバ名指示
4. 信頼される CA 指示

3.4.1.1 再ネゴシエーション指示

TLS セッション再ネゴシエーションは、攻撃者が TLS コネクションを形成し、選択的に内容を注入し、次に正当なクライアントからの新しい TLS コネクションにおいて接合するような標的サーバ上での攻撃に対して脆弱である。サーバは、攻撃者のネゴシエートされたセッションの再ネゴシエーションとして正当なクライアントの初期 TLS ハンドシェイクを取り扱う、したがって攻撃者によって送出された

19 タイムスタンプ値は、TLS において正しいものである必要はない。特段のより高レベルまたはアプリケーションプロトコルによって制限されなければ、任意の 4 バイトであればよい。

初期データが正当なクライアントからのものであると信じてしまう。セッション再ネゴシエーション拡張は、セッション接合またはセッション傍受等を防止するために定義される。拡張は、初期セッションネゴシエーションとセッション再ネゴシエーションを暗号学的に結合するような概念を使用する。

サーバは、[RFC5746]に従って]初期および続く再ネゴシエーションを実行し**なければならない**。

3.4.1.2 証明書状態要求

TLS サーバから TLS サーバ証明書の失効状態を受信したいとクライアントが望むとき、クライアントは、ClientHello メッセージ中に証明書状態要求拡張（Certificate Status Request (status_request) extension）を含める。status_request の受審に際して、サーバは、Certificate メッセージに続いて、CertificateStatus メッセージを直ちに送信することによって、その証明書と共に証明書状態を含め**なければならない**。拡張それ自体は拡張可能であるが、OCSP タイプの証明書状態のみは、[RFC6066]において定義されている。この拡張は、OCSP ステープリングとも呼ばれる。

3.4.1.3 サーバ名表示（Sever Name Indication：SNI）

複数の仮想的なサーバが同じネットワークアドレスに存在するかもしれない。サーバ名表示拡張は、接続しようとするアドレスにあるサーバをクライアントが指定できるようにする。サーバは、[RFC6066]で記述されるとおり ClientHello メッセージで受信されるサーバ名表示拡張を処理でき、応答し**なければならない**。

3.4.1.4 信頼される CA 表示

信頼される CA 表示（trusted_ca_keys）拡張は、所持する CA ルート鍵をクライアントが指定できるようにする。これはクライアントがメモリ制約されており、少ないが図のルート CA 鍵を所持するようなセッションにとって役に立つ。サーバは、[RFC6066]で記述されるとおり ClientHello メッセージで受信される信頼される CA 表示拡張を処理でき、応答し**なければならない**。

3.4.2 条件付き TLS 拡張

TLS サーバは、以下のパラグラフに記述されるような状況下において以下の TLS 拡張をサポートできるかもしれない：

1. supported Elliptic Curves TLS 拡張は、サーバが EC 暗号スイートをサポートする場合にサポートされ**なければならない**。
2. EC Point Format TLS 拡張は、サーバが EC 暗号スイートをサポートする場合、サポートされ**なければならない**。
3. Signature Algorithms TLS 拡張は、サーバが TLS 1.2 で動作するとき、サポートされ**なければならない**。
4. Multiple Certificate Status 拡張は、拡張がサーバ実装によってサポートされる場合、サポートされ**なければならない**。
5. Truncated HMAC 拡張は、サーバが制限されたデバイスクライアントと通信し、サーバ実装が可変長パディングをサポートしない場合、サポートされてもよい。

3.4.2.1 サポートされる Elliptic Curves

楕円曲線暗号スイートをサポートするサーバは、ClientHello メッセージにおいて受信された楕円曲線を処理でき**なければならない**。曲線 P-256 と P-384 がサポートされ**なければならない**。サーバは、[RFC4492]のセクション 5.1 に従ってこの拡張を処理し**なければならない**。

3.4.2.2 EC Point Format

EC 暗号スイートをサポートするサーバは、クライアントによる ClientHello メッセージにおいて受信された supported EC point format を処理できなければならない。サーバは、[RFC4492]のセクション 5.1 に従ってこの拡張を処理しなければならない。

EC 暗号スイートをサポートするサーバは、[RFC4492]のセクション 5.2 で記述されるとおり ServerHello メッセージにおいて supported EC point format についても送信できなければならない。

3.4.2.3 署名アルゴリズム

TLS 1.2 をサポートするサーバは、ClientHello メッセージにおける受信された署名アルゴリズム拡張の処理をサポートしなければならない。拡張、文法および処理規則は[RFC5246]のセクション 7.4.1.4.1、7.4.2 および 7.4.3 に記述されている。

3.4.2.4 Multiple Certificate Status

multiple certificate status 拡張は、TLS ハンドシェイクにおいてサーバによって提供されるすべての証明書の状態を要求できることをクライアントに許容することによってセクション 3.4.1.2 に記述された Certificate Status Request 拡張を改良する。サーバがサーバ証明書チェインにおけるすべての証明書の失効状態を返すとき、クライアントは、OCSP レスポンダ等の任意の失効サービスプロバイダに問い合わせる必要がない。この拡張は、[RFC6961]に文書化されている。この機能を持つようなサーバ実装は、この拡張をサポートするよう設定されなければならない。

3.4.2.5 Truncated HMAC

Truncated HMAC 拡張は、MAC タグとして使用する HMAC 出力の 80bits へのトランケーションを許容する。80-bit MAC タグは、[SP800-107]における推奨事項に適合するが、完全性アルゴリズムによって提供されるセキュリティを減じてしまう。MAC タグの偽造はオンライン攻撃であり、不正な MAC タグは観測されるとき TLS セッションは直ちに終了するので、この拡張をサポートすることにより導入されるリスクは低い。しかし、トランケートされた MAC タグは、[Paterson11]で記述された攻撃のため、可変長パディングと共に使用されてはならない。

3.4.3 推奨されない TLS 拡張

以下の拡張は、使用されるべきではない：

1. Client Certificate URL

Client Certificate URL 拡張は、相互認証中にサーバへ証明書を送信するよりむしろ、証明書を指し示す URL を送信することをクライアントに許容する。これは、制限されたクライアントと相互承認するのに非常に役立つ可能性がある。しかし、この拡張は、悪意のある目的でも使用される可能性がある。URL は、クライアントがサービス拒否攻撃を実行し、TLS サーバを攻撃者に変えたいような、無実のサーバに属しているかもしれない。この拡張をサポートするサーバは、証明書を検索する際にクライアントとしても動作する、ゆえに追加のセキュリティ上の懸念の対象となる。これらの理由より、Client Certificate URL 拡張は、サポートされるべきではない。しかし、政府機関がそのリスクが最小限であると決定し、この拡張が、クライアントが制限されたデバイスであるような環境で必要とされる場合、この拡張はサポートされてもよい。client certificate URL 拡張がサポートされる場合、サーバは上記および[RFC6066]のセクション 11.3 で記述されるセキュリティ上の懸念事項を低減するよう設定されなければならない。

3.5 クライアント認証

強い暗号学的クライアント認証が要求される場合、TLS サーバはクライアントを暗号学的に認証する[20]ためクライアント証明書を要求するような TLS プロトコルクライアント認証オプションを使用することができる。例えば、個人識別検証(Personal Identity Verification(PIV)) 認証証明書[FIPS201-1] (および対応するプライベート鍵) は、現場へのアクセス許可を持つ連邦政府職員および請負業者の強い認証のための適切なオプションを提供する。政府機関が PIV カードを最大限に活用することを位置づけられていることを保証するため、クライアント認証を実行するすべての TLS サーバは証明書ベースのクライアント認証をサポートし**なければならない**。

クライアント認証オプションは、サーバが X.509 パス検証メカニズムとトラストアンカのストアを実装することを要求する。これらのメカニズムの要求事項は、それぞれ、3.5.1 および 3.5.2 において規定される。暗号学的認証が実際に強い認証をもたらすことを保証するため、クライアント鍵は少なくとも 112 bits のセキュリティを含ま**なければならない**。セクション 3.5.3 はこの要求事項を実施するために間接的ではあるが、貢献できるようなメカニズムについて記述している。セクション 3.5.4 は、サーバヒントリストのクライアントの使用について記述している。

TLS サーバは、クライアント証明書が要求され、クライアントは適切な証明書を持っていないとき、致命的な「ハンドシェイク失敗」警告とともにコネクションを終了するように設定可能で**なければならない**。

3.5.1 パス検証

クライアント証明書は、[RFC5280]のセクション 6 で規定される証明書パス検証規則に従って検証され**なければならない**。さらに、証明書パスにおけるそれぞれの証明書の失効状態は、証明書失効リストまたはオンライン証明書状態プロトコル(OCSP)を用いて検証され**なければならない**。OCSP チェックは、[RFC6960]に従っていなければならない、また以下のオプションのうちの一つのみを使用する**べきである**：

- OCSP レスポンダはサーバによって信頼されている、即ち、OCSP レスポンダ公開鍵は、サーバのトラストアンカストアにおける公開鍵の一つと同じである；または
- OCSP レスポンダは状態がチェックされている証明書と同じ鍵を用いて署名されている；または
- OCSP レスポンダは、[RFC6960]で記述されるような指定された／代理の OCSP レスポンダによって署名され、OCSP レスポンダ証明書は、チェックされている証明書と同じ鍵を用いて署名されている。

失効情報は、セクション3.2.2で記述されるとおり、取得され**なければならない**。
サーバは、クライアント証明書が[RFC5280]のセクション 6 で規定される証明書パス検証規則を用いて信頼されるような証明書ポリシーを決定できなければならない。サーバとハックエンドアプリケーションは、証明書を受け入れまたは拒否するために、この決定を使用することができる。証明書ポリシーをチェックすることは、CA と登録システムおよびプロセスのセキュリティに関して、受け入れ可能な保

---

20 Certificate Verify メッセージは、署名機能を持つクライアント証明書を明示的に検証するために送信される。TLS 1.1 (および TLS 1.0) においてこのメッセージは、それ以前に来たすべてのハンドシェイクメッセージ上の署名を生成するために SHA-1 を使用する。[SP800-131A]では、2013 年以降はディジタル署名生成のために SHA-1 の使用が許容されないと述べている。衝突(collision) が密方にも拘らず、クライアントがハッシュを署名することによりそれ自身を認証するためにそのプライベート鍵を使用しなければならない。クライアントのランダムとサーバのランダムにより、サーバ、クライアント、または第三者は、将来のコネクションにおけるクライアントまたはサーバとしてなりすましするような衝突している一連のメッセージを使用することはできない。このメッセージ、それに先立つメッセージ、または後続のメッセージに対する任意の改変は、最終的にはコネクション失敗を引き起こすだろう。これらの理由により、SHA-1 は TLS 証明書検証メッセージにおけるディジタル署名生成には許容される。

証を持って発行されたようなクライアント証明書が受け入れられることをサーバに保証する。

すべての商用製品が上記のような公開鍵証明書パス検証と証明書ポリシー処理規則をサポートしてもいというわけではない。クライアント認証を実装するとき、連邦政府機関は、これらの要求事項を満たす商用製品を使用するか、またはこれらの要件を満たすような追加の商用製品を使用するかのいずれかをし**なければならない**。

サーバは、アクセス制御決定をサポートするためのアプリケーションに対して、クライアント証明書、およびクライアント証明書パスが有効であるような証明書ポリシーを提供でき**なければならない**。

3.5.2 トラストアンカストア

必要以上の数のトラストアンカが TLS アプリケーションにインストールされることは、これらのトラストアンカから放出されるすべての PKI に対してアプリケーションをさらす可能性がある。暴露を最小化する最良の方法は、クライアント公開鍵証明書認証で絶対に必要なトラストアンカストアにおけるトラストアンカのみを含めることである。

サーバは、サーバが TLS でクライアント認証をサポートするような場合に、クライアントを認証するために必須であるようなもののみのような、サーバが信用するようなトラストアンカのみを用いて設定され**なければならない**。これらのトラストアンカは、通常デフォルトサーバ上に含まれているかもしれないようなトラストアンカの小さなサブセットである。即ち、デフォルトのトラストアンカのセットが、それらの任意のものがクライアント認証に要求されるかどうかを決定するために検査され**なければならない**。いくつかの具体的な企業および／または PKI サービスプロバイダトラストアンカは、追加される必要があるかもしれない。

米国連邦政府環境において、ほとんどの場合に、連邦政府共通ポリシールートまたは政府機関ルート(連邦政府ブリッジ認証局とクロス証明書される場合)がクライアント証明書への証明書パスを構築するのに十分であるべきである。

証明書ベースのクライアント認証をサポートするような TLS サーバのシステム管理者は、クライアント証明書発行者の分析を実行し**なければならず**、サーバに要求される最小限のトラストアンカを決定するためのそのような情報を使用し**なければならない**。サーバは、それらのトラストアンカを含めるようにのみ設定され**なければならない**。

3.5.3 クライアント鍵長のチェック

クライアント公開鍵証明書で提示される鍵長とアルゴリズムが受け入れ可能かどうかをチェックするためのサーバの直接のメカニズムのみがサーバがクライアントの証明書における公開鍵とアルゴリズムを検査することである。間接的なメカニズムは、クライアント公開鍵証明書における証明書ポリシー拡張が使用された署名とハッシュアルゴリズムの最小限の暗号学的強度を示すことをチェックすることであり、サーバが証明書ポリシー処理とチェックを実行することである。標準ベースであるが商用的には幅広い普及を得ていないような、よりスケーラブルでより堅牢な代替策は、附属書 D に記述されている。サーバは、クライアント認証が実行される場合、クライアント鍵長をチェックし**なければならない**、またサーバ実装は、そうするためのメカニズムを提供する。サーバは、クライアントがマスタシークレットの生成に一時的鍵を使用する場合、クライアント公開鍵長についてもチェックし**なければならない**、またサーバ実装はそうするためのメカニズムを提供する。連邦政府機関は、クライアント鍵長をチェックするために、[SP800-131A]で提供される鍵サイズガイドラインを使用し**なければならない**。

3.5.4 サーバヒントリスト

クライアントは、クライアントの証明書パスがこれらのトラストアンカの一つで終端するかどうかを決定するため、CertificateRequest メッセージにおいてサーバより送られたトラストアンカの一覧を使用することができる。サーバによって送られた一覧は「ヒントリスト」として知られている。サーバとク

ライアントは異なる PKI ドメインにあるとき、信頼は、二つの PKI ドメイン間の直接的なクロス証明書経由で確立される(即ち、サーバ PKI ドメインとクライアント PKI ドメイン)または遷移的クロス証明(即ち、複数の PKI ドメイン間のクロス証明を通して) 経由で確立され、クライアントのトラストアンカがヒントリストに送信されていないので、クライアントはその証明書がサーバによって受け入れられないと誤って決定するかもしれない。この失敗を軽減するため、サーバは、加入者がサーバのクライアントである可能性があるような、さまざまな PKI のトラストアンカを、そのヒントリストに含むように維持し**なければならない**。代わりに、サーバは、クライアントが常に所持する証明書を提供できるように空のヒントリストを送信するよう設定される**べきである**。しかし、このリストは、サーバのトラストアンカストア[21]から区別され**なければならない**。言い換えると、サーバは、そのトラストアンカストアにサーバの PKI ドメインのトラストアンカとクライアント認証で直接信頼する必要があるドメインを追加のみするように継続しなければならない。サーバヒントリストとサーバ自体のトラストストアの間の違いは、以下のとおりである：1)ヒントリストは、クライアントとなるかもしれないものが信頼するかもしれないトラストアンカのリストである；また 2)サーバのトラストストアは、サーバが明示的に信頼するようなトラストアンカのリストである。

3.6 セッション再開

クライアントとサーバ間の初期のハンドシェイク中に、サーバは、セッション識別子(ID)を生成し、この値をハンドシェイク中にクライアントへ渡す。サーバとクライアントの両方がハンドシェイクの完了後に後で使用するためにセッション ID(鍵材料と暗号スイートと共に)を保存する。サーバがクライアントの要求時にセッションを再開したい場合、サーバは、元のセッション ID とハンドシェイクの開始時の暗号スイートを用いて応答する。サーバがセッションを再開したくないような事象において、サーバは新しいセッション ID を生成し、応答する。

通常のサーバ実装は、以前のセッションを再開することに合意可能である。これは、クライアントとサーバのみに知られているマスタシークレットとして、セキュアな利用モードであり、クライアント認証が要求される場合、初期のクライアント認証と結合される。しかし、コネクションセッションを初期化するたびにそれぞれのクライアントを認証するような要求事項がある場合、サーバは、セッションを再開する要求を無視するよう設定され**なければならず**、またハンドシェイク手順全体(クライアント認証を含めて)を実施するような新しいセッション ID を生成する。

3.7 圧縮方法

圧縮の使用は、攻撃者に圧縮ベースのサイドチャネルを用いた攻撃を実行可能とするかもしれない。これゆえに、TLS 圧縮を無効化するような、null 圧縮方法のみが使用される**べきである**。圧縮が使用される場合、[RFC3749]で定義される方法が使用され**なければならない**。提供されるクライアント追加が[RFC3943]での圧縮方法をサポートする場合、その方法が代わりに使用されてもよい。その他の圧縮方法は、使用**されてはならない**。圧縮方法推奨事項は、TLS 標準に基づいている。制限は、相互運用性を保証するために推奨される。

3.8 運用上の検討事項

上記セクションは、TLS 特有の機能を規定する。この機能は運用環境においてセキュリティを達成するために、必要であるが、十分ではない。

連邦政府機関は、TLS サーバが[SP800-53]等のその他の NIST ガイドラインで規定されるとおりの適切なネットワークセキュリティ保護を含むことを保証し**なければならない**。

---

21 サーバとクライアントのトラストアンカによっては、二つのリストが同一である可能性があるが、一般にいくつかのトラストアンカを持っている可能性があり、またまったく持っていいない可能性がある。

サーバは、セキュアなオペレーティングシステム[22]上で操作し**なければならない**。サーバが FIPS 140 レベル 1 暗号モジュールに信頼を置く場合、ソフトウェアとプライベート鍵はオペレーティングシステムの識別、認証およびアクセス制御メカニズムを用いて保護され**なければならない**。いくつかの高度に機微なアプリケーションにおいて、サーバのプライベート鍵は FIPS 140 レベル 2 またはより高度なハードウェア暗号モジュールを用いた保護を要求するかもしれない。

サーバと関連プラットフォームは、セキュリティパッチに関して最新を維持し**なければならない**。これは、製品ベンダによって送られてくる証明書のブラックリストを含めてさまざまなセキュリティの観点で重要である。証明書のブラックリストは、上流へさかのぼる CA 証明書またはクライアント証明書が無効であると宣言されているか、適切なセキュリティ対策と共に運用されていないとき、およびサーバが失効チェックを実行していないとき、最新の失効情報へアクセスしていないとき、または証明書が失効されていないときに、役に立つ。

3.9 サーバ推奨事項

このセクションは、セクション 3.1 からセクション 3.8 までの推奨事項を TLS サーバの選択、設定および維持のために要約したものを含む。

3.9.1 サーバ選択の推奨事項

以下の推奨事項の要約は、調達する TLS サーバ実装の選択を担当する個人のためのものである。TLS サーバ実装は、要求された機能を含むことなしに調達**されてはならない**。サーバ選択の推奨事項は以下のとおりである：

1. サーバ実装は、TLS バージョン 1.1 をサポートし**なければならない**。
2. サーバ実装は、TLS バージョン 1.2 をサポートする**べきである**。
3. サーバ実装は、TLS バージョン 1.0 をサポートしてもよい。
4. TLS バージョンネゴシエーションを不正確に実装したようなサーバ実装は、選択**されてはならない**。
5. サーバ実装は、パディングエラーを示すような bad_record_error エラーを使用し**なければならない**。
6. サーバ実装は、パディングエラーが存在するかにかかわらず MAC を計算し**なければならない**。
7. サーバ実装は、一定時間復号、またはほぼ一定時間復号をサポートする**べきである**。
8. サーバ実装は、アルゴリズムと鍵長の俊敏性をサポートするため、複数のサーバ証明書とそれらのプライベート鍵をサポートする**べきである**。
9. サーバ実装は、[SP800-90A]で規定された承認された乱数ビット生成器を使用し**なければならない**。
10. サーバ実装は、クライアントが証明書または受け入れ可能な証明書を持っていないとき、[致命的なハンドシェイク失敗」警告と共にコネクションを終了でき**なければならない**。
11. サーバ実装は、証明書失効リスト(CRL)またはオンライン証明書状態プロトコル(OCSP)、または両方をサポートするよう設定可能で**なければならない**。
12. サーバ実装は、セクション 3.5.1 のパス検証推奨事項をサポートするか、またはそれらをサポートするために追加され**なければならない**。
13. サーバは、アクセス制御決定をサポートするためにアプリケーションに対して、クライアント証明書、およびクライアント証明書パスが有効であるような証明書ポリシーを提供でき**なければならない**。

---

22 セキュアオペレーティングシステムは、以下の特徴を持つと共に使用する：アプリケーションとプロセスからのオペレーティングシステムの保護；アプリケーションとプロセスの間のオペレーティングシステム仲介の分離；利用者識別と認証；認証された利用者識別情報に基づくアクセス制御、およびセキュリティ関連アクティビティのイベントログ。

3.9.2 サーバのインストールと設定のための推奨事項

以下の推奨事項の要約は、TLS サーバ実装のインストールおよび初期設定を担当する個人のためのものである。TLS サーバ設定の推奨事項は、以下のとおりである：

1. バージョンサポート
   a. サーバは、TLS バージョン 1.1 をサポートするよう設定され**なければならない**。
   b. サーバは、TLS バージョン 1.2 をサポートするよう設定される**べきである**。
   c. サーバが政府専用アプリケーションをサポートする場合、TLS バージョン 1.0 をサポートするよう設定**されてはならない**。
   d. サーバがアプリケーションに接する市民または業界をサポートする場合、TLS バージョン 1.0 をサポートするよう設定されてもよい。
   e. TLS 1.0 がサポートされる場合、TLS 1.1 および 1.2 は TLS 1.0 よりも優先され**なければならない**。
   f. サーバは SSL 2.0 または SSL 3.0 をサポートするよう設定**されてはならない**。
2. 証明書
   a. サーバは、一つまたはそれ以上の公開鍵証明書と対応するプライベート鍵と共に設定され**なければならない**。
   b. サーバは、RSA 鍵暗号化証明書と共に設定され**なければならない**。
   c. サーバは、ECDSA 署名証明書または RSA 署名証明書と共に設定される**べきである**。
   d. サーバが RSA 署名証明書と共に設定されない場合、署名のためにスイート B に名前のある曲線を用いた ECDSA 署名証明書と ECDSA 証明書の公開鍵が使用される**べきである**。
   e. サーバは、自己署名証明書よりもむしろ、CA によって発行された証明書と共に設定され**なければならない**。
   f. サーバ証明書は、CRL または OCSP 応答のいずれかで失効情報を発行するような CA によって発行され**なければならない**。
   g. 失効情報の情報源は、相互運用性を促進するために適切な拡張において証明書の中に含まれ**なければならない**。
   h. すべてのサーバ証明書は、X.509 バージョン 3 証明書で**なければならない**。
   i. 証明書と署名に含まれる両方の公開鍵は、少なくとも 112 bits のセキュリティ強度を持た**なければならない**。さらに、一時的鍵は、マスタシークレットを確立するために使用されるとき、少なくとも 112 bits のセキュリティ強度を持た**なければならない**。
   j. 証明書は、セクション 3.2.1 で記述されるとおり、公開鍵と一貫するアルゴリズムで署名され**なければならない**。
   k. サーバは、サーバ認証拡張鍵用途拡張をサポートするよう設定される**べきである**。
   l. 政府機関特有のサーバ証明書プロファイル要求事項が無い場合、表 3-1 の証明書プロファイルがサーバ証明書に使用される**べきである**。
   m. サーバは、クライアント認証が使用されるとき、クライアント証明書の失効チェックを実行しなければならない。
      i. 失効情報は、セクション 3.2.2 で記述される一つまたはそれ以上の場所からサーバによって得られ**なければならない**。
      ii. サーバが現在の失効情報を取得できないとき、証明書を受け入れるか、拒否するかの決定は、政府機関のポリシーに従ってなされる**べきである**。
   n. 政府機関特有のポリシーが無い場合、連邦政府機関は、共通ポリシーを使用し**なければならない**。
3. 暗号サポート
   a. サーバは、データ機密性と完全性サービスのために設定され**なければならない**。
   b. サーバは、全体的に承認されたアルゴリズムから設定される暗号スイートのみをサポートするよう設定され**なければならない**。
   c. サーバは、以下の暗号スイートをサポートするよう設定され**なければならない**：
      TLS_RSA_WITH_3DES_EDE_CBC_SHA
      TLS_RSA_WITH_AES_128_CBC_SHA

d. サーバは、以下の暗号スイートをサポートするよう設定される**べきである**：
   TLS_RSA_WITH_AES_256_CBC_SHA
   TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
   TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
   TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
   TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
e. サーバが TLS バージョン 1.2 をサポートするよう設定される場合、サーバは、以下の暗号スイートをサポートするよう設定され**なければならない**：
   TLS_RSA_WITH_AES_128_GCM_SHA256
f. サーバが TLS バージョン 1.2 をサポートするよう設定される場合、サーバは、以下の暗号スイートをサポートするよう設定される**べきである**：
   TLS_RSA_WITH_AES_256_GCM_SHA384
   TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
   TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
   TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384
   TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
   TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256
g. サーバは、セクション 3.3.1 に記述されるとおり、その他の受け入れ可能な暗号スイートをサポートするようにし設定してもよい。
h. サーバは、少なくとも 112 bits のセキュリティ強度を提供する署名を含む有効な証明書を持つための暗号スイートのみをサポートし**なければならない**。
i. サーバは、セクション 3.3.1 または附属書 C にあらわれないような暗号スイートを使用して設定**されてはならない**。
j. RSA 証明書について、鍵用途拡張は、RSA を用いて鍵交換を実行するような暗号スイートの鍵暗号化を規定し**なければならない**、また鍵用途拡張は、ECDHE 鍵交換を用いる暗号スイートのディジタル署名を規定し**なければならない**。
k. サーバによって使用される暗号モジュールは、FIPS140 認証された暗号モジュールで**なければならない**。
l. 暗号スイートに含まれるすべての暗号モジュールは、乱数生成器と同様に認証の範囲内で**なければならない**。
m. 乱数生成器は、NIST 暗号アルゴリズム確認プログラム(CAVP)の下で[SP800-90A]に従ってテストされ、確認されなければならない、またこのテストの結果である合格は、暗号モジュールの FIPS 140 認証書上に示され**なければならない**。
n. 認証された乱数生成器は、サーバのランダム値の 28-byte ランダム値を生成するために使用され**なければならない**。
o. 認証された乱数生成器は、サーバランダム値の 4-byte タイムスタンプを生成するために使用される**べきである**。

4. 拡張
a. TLS サーバは、セクション 3.4.1 で記述されるとおり、以下の TLS 拡張をサポートし**なければならない**：
b. TLS サーバは、セクション[3.4.2 で述べられる条件が満たされるとき、セクション 3.4.2 で記述されるとおり、以下の TLS 拡張をサポートし**なければならない**：
c. Supported Elliptic Curves 拡張がサポートされるとき、曲線 P-256 と P-384 がサポートされ**なければならない**。
d. TLS サーバは、セクション 3.4.2 で述べられる条件が満たされるとき、セクション 3.4.2.5 で記述されるとおり、Truncated HMAC 拡張をサポートしれもよい。
e. TLS サーバは、Client Certificate URL 拡張をサポートする**べきではない**。
f. Client Certificate URL 拡張がサポートされる場合、サーバはセクション 3.4.3 で記述される攻撃を軽減するように設定され**なければならない**。
g. 可能であれば、サーバは、クライアントの一時的公開鍵が少なくとも 112bits のセキュリティ強度を提供できることを保証するためにマスタシークレットを確立するために使用されるク

ライアントの一時的公開鍵長をチェックしなければならない。

5. クライアント認証
   a. サーバがクライアント認証をサポートする場合、証明書ベースのクライアント認証をサポートしなければならない。
   b. 可能であれば、サーバは、クライアント公開鍵が少なくとも 112bits のセキュリティ強度を提供できることを保証するため、クライアント公開鍵長をチェックしなければならない。
   c. サーバは、クライアント証明書が要求されて、クライアントが適切な証明書を持っていないとき、致命的な「ハンドシェイク失敗」警告と共にコネクションを終了するよう設定されなければならない。
   d. クライアント認証が実行されるとき、サーバは、[RFC5280]のセクション 6 で規定される証明書パス検証規則に従ってクライアント証明書を検証しなければならない。
   e. サーバは、証明書パスにおけるそれぞれの証明書が証明書失効リスト(CRL)またはオンライン証明書状態プロトコル(OCSP)を用いて検証されなければならないように設定されなければならない。
   f. サーバが OCSP をサポートする場合、OCSP チェックは、[RFC6960]と一貫していなければならず、また本文書のセクション 3.5.1 に記述されるオプションの一つを用いるべきである。
   g. サーバは、[RFC5280]のセクション 6 で規定される証明書パス検証規則を用いることによってクライアント証明書が信頼されるような証明書ポリシーを決定できなければならない。
   h. サーバは、サーバが信頼するトラストアンカのみ、およびサーバが TLS でクライアント認証をサポートするような場合にクライアントを認証することを要求されるようなもののみ、と共に設定されなければならない。
   i. サーバのトラストアンカのデフォルトセットは、それらの任意のものがクライアント認証で要求されるかどうかを決定するために検査されなければならない。
   j. サーバは、クライアント認証が実行される場合クライアント鍵長をチェックしなければならない、またサーバ実装はそうするためのメカニズムを提供する。
      i. 連邦政府機関は、クライアント鍵長をチェックするために、[SP800-131A]で提供される鍵サイズガイドラインを使用しなければならない。
   k. サーバは、その加入者がサーバのクライアントとなる可能性があり、それらをヒントリストに含むような、さまざまな PKI のトラストアンカを維持するよう設定されなければならない。
      i. その代わりに、サーバは、クライアントが常に所持する証明書を提供できるように、空のヒントリストを送信するよう設定されるべきである。
   l. サーバは、サーバヒントリストは、サーバのトラストアンカストアから区別されなければならない。
   m. サーバは、サーバ PKI ドメインのトラストアンカおよびクライアント認証で直接信頼する必要がるようなドメインに、そのトラストアンカストアのみを追加し続けなければならない。
6. セッション再開
   a. コネクションセッションを開始する際にそれぞれのクライアントを認証するような要求事項がある場合、サーバは、セッションを再開する要求を無視して、処理を進めるために(クライアント認証を含めて)ハンドシェイク手順全体を実施するような新しいセッション ID を生成するよう設定されなければならない。
7. 圧縮方法
   a. サーバは、TLS 圧縮を無効化するような、null 圧縮方法のみをサポートするよう設定されるべきである。
   b. 圧縮が使用される場合、サーバは、[RFC3749]で定義される方法のみをサポートするよう設定されなければならない。
      i. 提供されるクライアント追加は、[RFC3943]の圧縮方法をサポートすると知られている場合、その方法は代わりに使用されてもよい。
   c. サーバは、その他のアッシュ方法をサポートするよう設定されてはならない。
8. 運用上の検討事項
   a. サーバは、セキュアなオペレーティングシステム上で動作しなければならない。

b. サーバが FIPS140 レベル 1 暗号モジュールに信頼を置いている場合、ソフトウェアとプライベート鍵はオペレーティングシステムの識別、認証およびアクセス制御メカニズムを用いて保護され**なければならない**。

3.9.3 サーバシステム管理者のための推奨事項

サーバシステム管理者とは、日々の TLS サーバを維持する責任を持つような個人である。

1. バージョンサポート
   a. システム管理者は、2015 年 1 月 1 日までに、TLS 1.2 をサポートするための移行計画を策定し**なければならない**。
2. 証明書
   a. システム管理者は、証明書の適切な情報源を識別するため、セクション 3.2.1 および 3.2.2 を使用しなければならない。
   b. システム管理者は、セクション 3.9.2 の証明書推奨事項に従って証明書をインストール、維持、および更新し**なければならない**。
3. 暗号サポート
   a. システム管理者は、セクション 3.9.2 の推奨事項に従って、機密性と完全性サービス設定を維持し**なければならない**。
4. クライアント認証
   a. 証明書ベースのクライアント認証をサポートするような TLS サーバのシステム管理者は、クライアント証明書発行者の分析を実行し**なければならず**、またはサーバに要求される最小限のトラストアンカを決定するためにその情報を使用し**なければならない**。
      i. サーバは、必要とされるトラストアンカの最小限のセットのみを含むのみとなるよう設定され**なければならない**。
5. 運用上の検討事項
   a. システム管理者は、TLS サーバがその他の NIST ガイドラインで規定されるとおり、適切なネットワークセキュリティ保護を含むことを保証し**なければならない**。
   b. サーバは、セキュアなオペレーティングシステム上で動作し**なければならない**。
   c. サーバが FIPS140 レベル 1 暗号モジュールに信頼を置く場合、システム管理者は、ソフトウェアとプライベート鍵がオペレーティングシステムの識別、認証およびアクセス制御メカニズムを用いて保護されることを保証し**なければならない**。
   d. システム管理者は、サーバと関連するプラットフォームがセキュリティパッチに関して最新に維持されることを保証しなければならない。

4 TLS クライアントの最小限の要求事項

本セクションは、これらのガイドラインを忠実に守るために TLS クライアントが満たさなければならないような要求事項の最小限のセットを提供する。要求事項は以下のセクションにおいて設定される：TLS プロトコルバージョンサポート；クライアント鍵と証明書；暗号サポート；TLS 拡張サポート；サーバ認証；セッション再開；圧縮方法；および動作上の系統事項。

具体的な要求事項は実装要求事項または設定要求事項のいずれかとして記述される。実装要求事項は、連邦政府機関が要求される機能を含まない TLS クライアント実装を調達**してはならない**ことを示す。設定要求事項は、システム管理者が徳知恵の機能が有効化され、または何らかの場合に、存在する場合は適切に設定されることを検証するよう要求されることを示す。

4.1 プロトコルバージョンサポート

クライアントは、TLS 1.1 をサポートするよう設定され**なければならない**、また TLS 1.2 をサポートするよう設定される**べきである**。クライアントは、必要な場合に、民間サービスとの通信を促進するため、TLS 1.0 をサポートするよう設定してもよい。TLS 1.0 がサポートされる場合、TLS1.1 および 1.2 の使用が TLS 1.0 よりも優先され**なければならない**。クライアントは、SSL バージョン 3.0 またはそれ以前をサポート**してはならない**。政府機関は、2015 年 1 月 1 日までに TLS 1.2 をサポートするための移行計画を策定し**なければならない**。

4.2 クライアント鍵と証明書

4.2.1 クライアント証明書プロファイル

クライアント認証が必要とされるとき、クライアントは、このセクションで提示される推奨事項を遵守する証明書と共に設定され**なければならない**。クライアント証明書は、システム上で設定されるか、または外部デバイス上に配置されてもよい(例．PIV カード)。この仕様について、TLS クライアント証明書は、X.509 バージョン 3 証明書で**なければならない**；証明書に含まれる公開鍵と署名の両方が少なくとも 112bits のセキュリティ強度を持っていて**なければならない**。証明書は、公開鍵と一致する暗号アルゴリズムで署名され**なければならない**。

- RSA (署名)、ECDSA、またはDSA公開鍵を含む証明書は、それぞれ同じ署名アルゴリズムで署名され**なければならない**；
- Diffie-Hellman証明書を含む証明書は、DSAで署名され**なければならない**；そして
- ECDH公開鍵を含む証明書は、ECDSAで署名され**なければならない**。

拡張された鍵用途拡張は、証明書の鍵が使用されるような操作を制限する。特にクライアント認証のための鍵用途拡張がある。拡張された鍵用途拡張は、サーバがクライアント証明書として証明書を受け入れることを保証する。拡張された鍵用途拡張は、証明書が、コード署名等のその他の目的で使用されるべきでないことについても示している。クライアント証明書は、クライアント認証鍵目的オブジェクト識別子 [23]を規定するような拡張された鍵用途拡張を含む**べきである**。

クライアント証明書プロファイルは、表 4-1 で列挙されている：TLS クライアント証明書プロファイル。政府機関特有のクライアント証明書プロファイルが無いとき、このプロファイルがクライアント証明書のために使用される**べきである**。

ECDH について、アルゴリズム OID と署名 OID は ECDSA のものと同一であることに注意すること。相互運用性の理由のため、アルゴリズム OID は変更されず、鍵用途拡張は公開鍵が鍵共有または署名

---

23 いくつかの実装で extended key usage 拡張がないものが、証明書で特別に示されていないにもかかわらず、コード署名等で特別に許可されたものとして解釈されるように知られている。

検証のために使用されるかどうかを決定する。

Field フィールド	Critical	Value 値	Description 説明
Version バージョン	N/A	2	Version 3 バージョン 3
Serial Number シリアル番号	N/A	Unique positive integer ユニークな正の整数	Must be unique ユニークでなければならない
Issuer Signature Algorithm 発行者署名アルゴリズム	N/A	*Values by certificate type:証明書タイプによる値*	
		sha256WithRSAEncryption {1 2 840 113549 1 1 11}, or stronger	RSA signature certificate RSA 署名証明書
		ecdsa-with-SHA256 {1 2 840 10045 4 3 2}, or stronger	ECDSA signature certificate, ECDH certificate ECDSA 署名証明書、ECDH 証明書
		id-dsa-with-sha256 {2 16 840 1 101 3 4 3 2}, or stronger	DSA signature certificate, DH certificate DSA 署名証明書、DH 証明書
Issuer Distinguished Name 発行者識別名	N/A	Unique X.500 Issuing CA DN ユニークな X.500 発行 CA 識別名	Single value shall be encoded in each RDN. All attributes that are of directoryString type shall be encoded as a printable string. 一つの値がそれぞれの関連識別名(RDN)においてエンコードされなければならない。directoryString タイプであるすべての属性が PrintableString としてエンコードされなければならない。
Validity Period 有効期間	N/A	3 years or less 3 年以下	Dates through 2049 expressed in UTCTime UTCTime で表現された 2049 年までの日付
Subject Distinguished Name サブジェクト識別名	N/A	Unique X.500 subject DN per agency requirements 政府機関要求事項ごとのユニークな X.500 サブジェクト識別名	Dates value shall be encoded in each RDN. All attributes that are of directoryString type shall be encoded as a printable string. CN={Host IP Address | Host DNS Name}
Subject Public Key Information サブジェクト公開鍵情報	N/A	*Values by certificate type:証明書タイプによる値*	
		rsaEncryption (1 2 840 113549 1 1 1}	RSA key encipherment certificate, RSA signature certificate 2048-bit RSA key modulus, or other approved lengths as defined in [FIPS186-4] (訳注:原文は FIPS168-4 だが、186-4 の間違い)and [SP800-57p1]] Parameters: NULL. RSA 鍵暗号化証明書､RSA 署名証明書 2048-bitRSA 鍵モジュラス、または[FIPS186-4]および[SP800-57p1]で定義されたとおりのその他の承認された長さ パラメータ：なし
		ecPublicKey {1 2 840 10045 2 1}	ECDSA signature certificate, or ECDH certificate Parameters: namedCurve OID for names curve specified in FIPS 186-4. The curve shall be P-256 or P-384 SubjectPublicKey: Uncompressed EC Point. ECDSA 署名証明書、または ECDH 証

Field フィールド	Critical	Value 値	Description 説明
			明書 パラメータ：FIPS 186-4 で規定された曲線目の namedCurve OID。曲線は P-256 または P-384 でなければならない。 SubjectPublicKey:非圧縮の EC Point。
		id-dsa {1 2 840 10040 4 1}	DSA signature certificate Parameters: p, g, q DSA 署名証明書 パラメータ： p, g, q
		dhpublicnumber {1 2 840 10046 2 1}	DH certificate Parameters: p, g, q DH 証明書 パラメータ：p, g, q
Issuer's Signature 発行者の署名	N/A	*Values by certificate type:証明書タイプによる値*	
		sha256WithRSAEncription {1 2 840 113549 1 11}, or stronger	RSA key encipherment certificate, RSA signature certificate RSA 鍵暗号化証明書､RSA 署名証明書
		ecdsa-with-SHA256 {1 2 840 10045 4 3 2}, or stronger	ECDSA signature certificate, ECDH certificate ECDSA 署名証明書、ECDH 証明書
		id-dsa-with-sha256 {2 16 840 1 101 3 4 3 2}, or stronger	DSA signature certificate, DH certificate DSA 署名証明書、DH 証明書
Extensions			
Authority Key Identifier オーソリティ鍵識別子	No	Octet string	Same as subject key identifier in Issuing CA certificate Prohibited: Issuer DN, Serial Number tuple 発行 CA 証明書におけるサブジェクト鍵識別子と同じ 禁止：発行者識別名、シリアル番号タプル(組)
Subject Key Identifier サブジェクト鍵識別子	No	Octet String	Same as in PKCS-10 request or calculated by the Issuing CA PKCS-10 リクエスト内と同じまたは発行 CA により計算されたものと同じ
Key Usage 鍵用途	Yes	digitalSignature	RSA certificate, DSA certificate, ECDSA certificate RSA 証明書、DSA 証明書、ECDSA 証明書
		keyAgreement	ECDH certificate, DH certificate ECDH 証明書、DH 証明書
Extended key Usage 拡張鍵用途	No	id-kp-clientAuth {1 3 6 1 5 5 7 3 2}	Required 必須
		anyExtendedKeyUsage {2 5 29 37 0}	Prohibited 禁止 [24]
			Prohibited: all others unless consistent with key usage extension 禁止：鍵用途拡張と一貫しないその他すべて

24 いくつかの実装における anyExtendedKeyUsage {2 5 29 37 0} の存在は証明書で特別に示されていないにもかかわらず、コード署名等の特別に許可されたものとして解釈されるように知られている。

Field フィールド	Critical	Value 値	Description 説明
Certificate Policies 証明書ポリシー	No	Per agency X.509 certificate policy	
Subject Alternative Name サブジェクト別名	No	RFC 822 e-mail address, Universal Principal Name (UPN), DNS Name, and/or others	Optional オプション
Authority Information Access オーソリティ情報アクセス	No	id-ad-caIssuers	Required. Access method entry contains HTTP URL for certificates issued to Issuing CA 必須。アクセス方法入力には発行 CA へ発行された証明書への HTTP URL が含まれる
		id-ad-ocsp	Optional. Access method entry contains HTTP URL for the Issuing CA OCSP Responder オプション。アクセス方法入力には発行 CA の OCSP レスポンダの HTTP URL が含まれる
CRL Distribution Points CRL 配付ポイント	No	See comments	Optional. HTTP value in distributionPoint field pointing to a full and complete CRL. Prohibited: reasons and cRLIssuer fields, and nameRelativetoCRLIssuer CHOICE オプション。すべての CRL および完全な CRL を指す distributionPoint フィールドにおける HTTP 値 禁止：reasons and cRLIssuer fields、nameRelativetoCRLIssuerCHOICE

複数のクライアント証明書は、TLS サーバの要求事項を満たすように存在するかもしれない。TLS クライアント(例．ブラウザ）は、証明書のリストから選択するよう利用者に依頼するかもしれない。Extended Key Usage(EKU)拡張はこの要求を取り除くかもしれない。

クライアント証明書は、また、セクション 3.5.4 で記述されるとおり、サーバによって送信されるヒントリストにおけるトラストアンカの一つへのパスを構築する能力に基づいて TLS クライアントによってフィルタされる。

4.2.2 サーバ証明書の失効状態情報の取得

クライアントは、サーバ証明書の失効チェックを実行し**なければならない**。失効情報は、以下のロケーションの一つからクライアントによって取得されることが可能である。

1. サーバの CertificateStatus メッセージ[RFC6066]、[RFC6961]における OCSP レスポンス。
2. クライアントのローカル証明書ストアにおける証明書失効リスト(CRL)または OCSP [RFC6960] レスポンス；
3. ローカルに設定された OCSP レスポンダからの OCSP レスポンス；
4. サーバ証明書のオーソリティ情報アクセス拡張の OCSP フィールドで識別された OCSP レスポンダロケーションからの OCSP レスポンス；
5. サーバ証明書の CRL 配付ポイント拡張からの CRL。

サーバが失効状態を提供せず、ローカル証明書ストアが現在のまたは説得力のある CRL または OCSP レスポンスを持たず、かつ OCSP レスポンダと CRL 配付ポイントが TLS セッション確立時に利用できない、またはアクセスできないとき、クライアントは、コネクションを終了するか、あるいは失効し

たかまたは危殆化した可能性のある証明書を受け入れるかのいずれかを行うこと。この状況で証明書を受け入れるかまたは拒否するかの決定は、政府機関のポリシーに従ってなされる**べきである**。

失効チェックの代わりに役に立つ可能性のあるその他の新出の概念が附属書 D でさらに議論される。

4.2.3 クライアント公開鍵証明書の保証

クライアント公開鍵証明書は、セクション 3.5.1 で記述されるとおりのクライアント公開鍵証明書を発行するために使用されるポリシー、手順およびセキュリティ管理策に基づいてサーバによって信頼されてもよい。例えば、個人識別検証(PIV) [FIPS201-1]の実装が連邦政府機関においてさらに確立されてくるので、これらのガイドラインは、PIV 認証証明書が連邦政府職員および長期間請負要因の認証のための基準となることを推奨する。外部の利用者等、PIV カードを持っていない利用者のため、受け入れる一連の証明書ポリシーは、アプリケー所によって要求される保証レベルに基づき、[SP800-63]の附属書 B で規定されるとおり決定されるべきである。PIV 認証証明書ポリシーは、[COMMON]で定義され、PIV-I 認証証明書ポリシーは、[FBCACP]で定義される。サーバ側サプリケーションの要求事項に依存して、[COMMON]で定義されるその他の証明書ポリシーは、受け入れ可能であるのかもしれない。受け入れ可能な証明書ポリシーに関するガイダンスは、本ガイドラインの適用範囲外である。

4.3 暗号サポート

4.3.1 暗号スイート

TLS クライアントの受け入れ可能な暗号スイートは、TLS サーバのそれと同じである。汎用暗号スイートは、セクション 3.3.1 に列挙されており、事前共有鍵環境の暗号スイートは附属書 C に列挙されている。一時的鍵がマスタシークレットを確立するために使用されるとき、それぞれの一時的鍵ペア(即ち、サーバの一時的鍵ペアとクライアントの一時的鍵ペア) は、少なくとも 112 bits のセキュリティ強度を持っていなければならない。

クライアントは、セクション 3.3.1 または附属書 C に列挙された暗号スイート以外のものを使用するよう設定されるべきではない。

CBC モードに対する攻撃を軽減するため、TLS 実装は、パディングエラーを示すために bad_record_mac エラーを使用しなければならない。実装は、パディングエラーが存在するかどうかにかかわらず MAC を計算しなければならない。TLS 実装は、一定時間復号、またはほぼ一定時間復号をサポートするべきである。

4.3.2 認証された暗号

クライアントは、セクション 3.3.2 のサーバのために記述されるとおり、認証された暗号を使用しなければならない。

認証された乱数生成器は、クライアントランダム値の 28byte ランダム値を生成するために使用されなければならない。認証された乱数生成器は、クライアントランダム値の 4-byte タイムスタンプを生成するために使用されなければならない。

4.4 TLS 拡張サポート

4.4.1 必須の TLS 拡張

クライアントは、以下の拡張をサポートしなければならない：

1. Renegotiation Indication 再ネゴシエーション指示
2. Sever Name Indication サーバ名指示

4.4.1.1 再ネゴシエーション指示

再ネゴシエーション指示拡張は、セクション 3.4.1.1 で記述されるとおりこれらのガイドラインによって要求される。クライアントは、[RFC5746]に従って初期のおよび後続の再ネゴシエーションを実行し**なければならない**。

4.4.1.2 サーバ名指示

サーバ名指示拡張は、セクション 3.4.1.3 で記述される。クライアントは、[RFC6066]で記述されるとおり、ClientHello メッセージでこの拡張を含むことができ**なければならない**。

4.4.2 条件付き TLS 拡張

TLS クライアントは、記述された環境下で、以下の TLS 拡張をサポートする：

1. Supported Elliptic Curves TLS 拡張は、クライアントが EC 暗号スイートをサポートする場合、サポートされ**なければならない**。
2. EC Point Format TLS 拡張は、クライアントが EC 暗号スイートをサポートする場合、サポートされ**なければならない**。
3. Signature Algorithm TLS 拡張は、クライアントが TLS 1.2 で動作するとき、サポートされ**なければならない**。
4. Certificate Status Request 拡張は、クライアントが失効情報を取得できないとき、サポートされ**なければならない**。
5. Multiple Certificate Status 拡張は、その拡張がクライアント実装によってサポートされるとき、サポートされ**なければならない**。
6. Trusted CA Indication 拡張は、少ない数の CA ルート鍵しか保存できないようなメモリ制約のあるデバイス上で動作するようなクライアントによってサポートされる**べきである**。
7. Truncated HMAC 拡張は、可変長パディングがサポートされないとき、制約のあるデバイス上で動作するクライアントによってサポートされてもよい。

4.4.2.1 サポートされる楕円曲線

EC 暗号スイートをサポートするようなクライアントは、[RFC4492]のセクション 5.1 に従って ClientHello メッセージでサポートされる楕円曲線を列挙でき**なければならない**。

4.4.2.2 EC ポイントフォーマット

EC 暗号スイートをサポートするクライアントは、[RFC4492]のセクション 5.1 に従って、ClientHello メッセージのサポートされる EC ポイントフォーマットを規定でき**なければならない**。

EC 暗号スイートをサポートするクライアントは、[RFC4492]のセクション 5.2 に記述されるとおり、ServerHello メッセージで受信された EC ポイントフォーマットの少なくとも一つ[25]の処理をサポートし**なければならない**。

---

25 [RFC4492]のセクション 5.1.2 および 5.2 で記述されるとおり、非圧縮のポイントフォーマットがサポートされなければならない。

4.4.2.3 署名アルゴリズム

TLS 1.2 をサポートするクライアントは、ClientHello メッセージでこの拡張において受け入れ可能なハッシュと署名アルゴリズムペアを行使できなければならない。拡張、その文法、および処理規則は、[RFC5246]のセクション 7.4.1.4.1、7.4.4、7.4.6 および 7.4.8 で記述される。

4.4.2.4 証明書状態要求

クライアントが TLS サーバから TLS サーバ証明書の失効状態を受信したいとき、クライアントは、ClientHello メッセージの「status_request」拡張を含めなければならない。

4.4.2.5 複数の証明書状態

multiple certificate status 拡張は、セクション 3.4.2.4 で記述されている。この拡張は、TLS ハンドシェイクでのサーバより提供されるすべての証明書の状態を要求することをクライアントに許可することによってセクション 3.4.1.2 で記述された Certificate Status Request 拡張を改良する。この拡張は、[RFC6961]で文書化されている。

この機能を持つクライアント実装は、この拡張をサポートするよう設定されなければならない。

4.4.2.6 信頼される CA 指示

クライアントは、[RFC6066]で記述されるとおり、ClientHello メッセージの信頼される CA 指示(trusted_ca_keys)拡張を含むことができるべきである。

4.4.2.7 Truncated HMAC

Truncated HMAC 拡張は、セクション 3.4.2.5 で記述される。制約されるデバイス上で動作するクライアントは、この拡張をサポートしてもよい。Truncated HMAC 拡張は、[Paterson11]で記述された攻撃のため、可変長パディングと共に使用されてはならない。

4.4.3 推奨されない TLS 拡張

以下の拡張は、使用されるべきでない：

1. クライアント証明書 URL

この拡張の使用を推奨しない理由は、セクション 3.4.3 で見つけることができる。

4.5 サーバ認証

クライアントは、適切なトラストアンカがストアに存在しない場合、クライアントストアのトラストアンカの少なくとも一つと TLS ハンドシェイクで提示されたサーバ証明書の証明パスを構築できなければならない。クライアントは、証明パスを構築するために以下の資源のすべてまたはサブセットを使用してもよい：ローカル証明書ストア、ハンドシェイク中にサーバから受信された証明書、LDAP、さまざまな CA 証明書での Subject Information Access 拡張の CA Repository フィールドで宣言された資源、およびさまざまな証明書における Authority Information Access 拡張の CA Issuers フィールドで宣言された資源。

4.5.1 パス検証

クライアントは、[RFC5280]のセクションで規定された証明書パス検証規則に従ってサーバ証明書を検

証し**なければならない**。さらに、証明パスにおけるそれぞれの証明書の失効状態は、証明書失効リスト(CRL)またはオンライン証明書状態プロトコル(OCSP)を用いてチェックされ**なければならない**。OCSPチェックは、[RFC6960]に適合し**なければならず**、また以下のオプションの一つだけを使用する**べきである**：

- OCSPレスポンダはクライアントによって信頼される、即ち、OCSPレスポンダ公開鍵は、クライアントのトラストアンカストアにおいて公開鍵の一つと同じものである；または
- OCSPレスポンスは、状態がチェックされる証明書のものと同じ鍵を用いて署名される；または
- OCSPレスポンスは、[RFC6960]で記述されるとおり指定された／代理のＯＣＳＰレスポンダによって署名される、またＯＣＳＰレスポンダは状態がチェックされる証明書のものと同じ鍵を用いて署名される。

失効情報は、セクション 4.2.2 で記述されるとおり取得され**なければならない**。

すべての商用製品が、上記の公開鍵証明書パス検証と証明書ポリシー処理規則をサポートするわけではない。特にいくつかの例での失効チェックは、利用可能でないかもしれない、またはクライアントは、最新の失効情報がアクセスできない場合に、サーバ証明書を受け入れてしまうかも知れない。同様に、何らかのクライアントは、受け入れ可能な証明書ポリシーまたはポリシー要求およびポリシーマッピングの阻止のための初期値に関連する入力を提供できない。十分に証明書ポリシーを理解しているクライアントの欠如において、連邦政府機関は、サーバ証明書が適切な配慮と共に発行されている場合、デバイスに対してその他のメカニズムを用いてもよい。

すべてのクライアントが名前制約チェックをサポートするわけではない。連邦政府機関は不正な証明書が適切に拒否されるような保証を得るため、名前制約チェックを実行するようなクライアントのみを調達し**なければならない**。代替案として、連邦政府機関は、附属書Ｄで議論される機能の一つまたは複数を使用するようなクライアントを調達してもよい。

クライアントは、パス検証が失敗した場合、TLS コネクションを終了し**なければならない**。

連邦政府機関は、クライアント TLS 要求で提示されるか、サーバ証明書の subject alternative name 拡張に含まれる DNS 名または IP アドレスと一致するかのいずれかであるような DNS 名または IP アドレスをチェックするようなクライアントのみを使用し**なければならない**。クライアント TLS 要求で提示される名前がサーバ証明書の subject alternative name 拡張にない場合のみ、クライアントはサブジェクトの識別名 (特に、コモン名属性タイプ) が要求された名前を含むかどうかを決定するため、サーバ証明書の subject distinguished name フィールドをチェックし**なければならない**。クライアントは、名前チェックが失敗したバイ、TLS コネクションを終了し**なければならない**。

4.5.2 トラストアンカストア

TLS クライアントに過度の数のトラストアンカがインストールされることは、クライアントが盗聴される機会を増加させる可能性がある。トラストアンカの数が増加するにつれ、クライアントが信頼する CA の数が増加し、これらの CA の一つ、または登録システム又はプロセスが TLS サーバ証明書発行に際して危殆化する機会も増加する。最小限の場合、連邦政府機関が信頼する当事者は、一つのトラストアンカを持つことができる：政府機関のレガシーなトラストアンカまたは共通ポリシートラストアンカ。

連邦政府機関は、さまざまなクライアントマシンにおいてトラストアンカを決定するために商用のウェブサイトをアクセスすることに関連するリスクと必要性の間のトレードオフを実行しなければならない。連邦政府機関は、中央管理アプリケーションを通してこのトラストアンカストアを管理しなければならない。連邦政府機関のシステムとクライアントは、トラストアンカの更新が適切な政府機関セキュアリティ承認を要求する特権システム管理者機能であるように設定されなければならない。

クライアント証明書選択とセクション 3.5.4 で記述されたクライアントエンドでのパス構築問題を軽減

するため、クライアントは、クロス証明経由で検証可能なさまざまなCA証明書をクライアントのトラストアンカに追加してはならない。これらの証明書の直接的な信頼は、これらのトラストアンカの失効または危殆化を含めて、それに限定はされないが、さまざまな状況に過度にクライアントをさらしてしまう可能性がある。直接的な信頼は、トラストアンカの追加と削除を広めるため、クライアント上での運用上およびセキュリティ上の負荷についても増加させる。それよりはむしろ、クライアントは、セクション 3.5.4 で議論されるとおり、過密またはヒントリストを提供しないサーバに信頼を置かなければならない。

4.5.3 サーバ鍵長のチェック

サーバ公開証明書で提示される鍵長が受け入れ可能であるかどうかをクライアントチェックするための直接的なメカニズムのみは、クライアントが証明書におけるサーバ公開鍵を検査することである。間接的なメカニズムは、サーバ公開鍵証明書の証明書ポリシー拡張が使用される署名およびハッシュアルゴリズムの最小限の暗号強度を示すことをチェックすることおよびクライアントが証明書ポリシー処理およびチェックを行うことである。標準ベースのよりスケーラブルでより堅牢な代替手段は、附属書 D で記述される。クライアントは、クライアント実装がそのようにするメカニズムを提供する場合、サーバ鍵長をチェックし**なければならない**。クライアントは、サーバがマスタシークレットの生成のために一時的鍵を使用する場合、サーバ公開鍵長についてもチェックし、またクライアント実装はこれを行うためのメカニズムを提供する。

各々の書込み鍵長は、ネゴシエートされた暗号スイートによって決定される。共有されたセッション鍵長の制約は、鍵長要求事項を満たすような暗号スイートのみをサポートするようにクライアントを設定することによって実施可能である。

4.5.4 利用者インタフェース

TLS クライアントがブラウザであるとき、ブラウザインタフェースは、TLS セッションが有効であるかどうかを決定するために使用されることが可能である。TLSセッションが有効であることの指示はブラウザによって変わる。指示の例には、URL バーのパッドロック、またはURL バーの異なる色を含む。ブラウザ等のいくつかのクライアントは、サーバ証明書のさらなる調査とロック(またはその他の表示器)上でクリックすることによるネゴシエートされたセッションパラメータを許容してもよい。利用者は、TLSセッションが実施されることを保証するための表示器の存在についてのインタフェースを検査する**べきである**、また表示されたウェブサイトを利用者が訪問しようとすることを保証するためにウェブサイトの URL を視覚的にも検査する**べきである**。利用者は URL が正当であるようにあらわされることが可能であるが、まだ有効ではないことを知っている**べきである**。例えば、数字の「1」と文字の「l」は全く同じよう、または人の目には同じに見える可能性がある。利用者が正しく見えるようなURL へナビゲートする場合、ブラウザソフトウェアはサーバ証明書の DNS 名を用いて要求された URL と照合することによって、これらの脅威を打倒することが可能である。

クライアント認証鍵は、クライアントの外(例．PIV カード)にあるかもしれない。利用者**は、クライアントの外のクライアント認証鍵を保護するためのポリシーと手順に従わ**なければならない。

4.6 セッション再開

クライアントは、セクション 3.6 で記述されるようなサーバと同じセッション再開推奨事項に従わ**なければならない**。

4.7 圧縮方法

クライアントは、セクション 3.7 で記述されるようなサーバと同じセッション再開推奨事項に従わ**なければならない**。

4.8 運用上の検討事項

クライアントと関連するプラットフォームは、セキュリティパッチに関して最新に維持されなければならない。これは、製品ベンダによって送られる証明書のブラックリストを含めて、セキュリティ上のさまざまな観点から重要である。証明書のブラックリストは、上流に向けた CA 証明書またはサーバ証明書が無効であると、または適切なセキュリティ手段を用いて運用されていないと宣言されており、かつクライアントが失効チェックを実行していない、最新の失効情報へアクセスしていない、または証明書が失効されているときに、役に立つ。

TLS 保護されたデータがクライアントで一度受信され、クライアントシステムの TLS レイヤによって復号され認証されると、暗号化されていないデータがクライアントプラットフォーム上のアプリケーションに対して利用可能となる。

これらのガイドラインは、クライアントマシン上に存在するクライアントクレデンシャルの誤使用または暴露に対する脅威についても軽減しない。これらのクレデンシャルは、クライアント認証、またはサーバ側アプリケーションへの認証のためのその他のクレデンシャル(例. ワンタイムパスワード(OTP)または利用者 ID とパスワード)で使用されるプライベート鍵を含んでいるかもしれない。

これらの理由について、TLS の利用は、適用可能な連邦政府情報処理標準および NIST Special Publication で記述されるとおり、コンピュータシステムとアプリケーションを保護するための、適切なセキュリティ手段をクライアントが使用する必要性を取り除かない。利用者は、政府機関および管理者の指示に従ってクライアントシステムを操作しなければならない。

4.9 クライアント推奨事項

本セクションは、TLS クライアントの選択、設定、維持、および使用のため、セクション 4.1 からセクション 4.8 までの要約された推奨事項を含んでいる。

4.9.1 クライアント選択の推奨事項

以下の推奨事項の要約は、TLS クライアント実装を調達のために選択することを担当する個人のためのものである。TLS クライアントは、それらが要求される機能を含むことなしに調達されてはならない。クライアント選択のための推奨事項は以下のとおりである：

1. クライアント実装は、TLS バージョン 1.1 をサポートしなければならない。
2. クライアント実装は、TLS バージョン 1.2 をサポートするべきである。
3. クライアント実装は、TLS バージョン 1.0 をサポートしてもよい。
4. クライアント実装は、TLS 1.1 および TLS 1.2 を TLS 1.0 よりも優先するように設定可能でなければならない。
5. クライアント実装は、パディングエラーを示すために bad_record_amc を使用しなければならない。
6. クライアント実装は、パディングエラーが存在するかどうかに拘わらず、MAC を計算しなければならない。
7. クライアント実装は、一定時間復号、またはほぼ一定時間復号をサポートするべきである。
8. クライアント実装は、client authentication extended key usag 拡張をサポートしなければならない。
9. クライアント実装は、許可されない証明書が適切に拒否されることを保証するため名前制限チェックをサポートしなければならない。
10. クライアント実装は、クライアント TLS 要求に存在する DNS 名または IP アドレスがサーバ証明書の subject distinguished name フィールドまたは subject alternative name 拡張に含まれる名前または IP アドレスと一致することをチェックしなければならない。
11. クライアント実装は、パス検証が失敗する場合、TLS コネクションを終了しなければならない。

4.9.2 クライアントのインストールと設定のための推奨事項

以下の推奨事項の要約は、TLS クライアント実装のインストールおよび初期設定を担当する個人のためのものである。TLS クライアント設定のための推奨事項は、以下のとおりである：

1. バージョンサポート

    a. クライアントは、TLS バージョン 1.1 をサポートするように設定され**なければならない**。
    b. クライアントは、TLS バージョン 1.2 をサポートするように設定される**べきである**。
    c. クライアントは、TLS バージョン 1.0 をサポートするように設定されてもよい。
    d. TLS バージョン 1.0 がサポートされる場合、クライアントは、TLS1.0 よりも TLS1.1 および TLS1.2 が優先されるように設定され**なければならない**。
    e. クライアントは、SSL バージョン 3.0 またはそれ以前をサポートするよう設定**されてはならない**。

2. 証明書

    a. すべてのクライアント証明書は、X.509 バージョン 3 証明書で**なければならない**。

    b. 証明書に含まれる公開鍵と署名の両方は、少なくとも 112 bits のセキュリティ強度を持っていなければならない。さらに、マスタシークレットを確立するために使用される場合、一時的鍵は少なくとも 112 bits のセキュリティ強度を持っていなければならない。

    c. 証明書は、セクション 4.2.1 で記述されるとおり、公開鍵を一貫したアルゴリズムを用いて署名され**なければならない**。

    d. クライアント証明書は、クライアント認証鍵目的オブジェクト識別子を規定するような extended key usage 拡張を含む**べきである**。

    e. 政府機関特有のクライアント証明書プロファイルが無い場合、表 4-1 のプロファイルがクライアント証明書に使用される**べきである**。

    f. クライアントは、セクション 4.2.2 で記述されるとおり、サーバ証明書の失効チェックを実行し**なければならない**。

        i. クライアントが現在の失効情報を取得できないとき、証明書を受け入れるか拒否するかの決定は政府機関のポリシーに従ってなされる**べきである**。

3. 暗号サポート

    a. クライアントは、以下の暗号スイートをサポートするよう設定され**なければならない**：
        TLS_RSA_WITH_3DES_EDE_CBC_SHA
        TLS_RSA_WITHAES_128_CBC_SHA
    b. クライアントは、以下の暗号スイートをサポートするよう設定される**べきである**：
        TLS_RSA_WITH_AES_256_CBC_SHA
        TLS_ECDHE_ECDSA_WITH_3DES_EDE_CBC_SHA
        TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA
        TLS_ECDHE_RSA_WITH_3DES_EDE_CBC_SHA
        TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA
    c. クライアントが TLS1.2 をサポートするよう設定される場合、クライアントは、以下の暗号スイートをサポートするよう設定され**なければならない**：
        TLS_RSA_WITH_AES_128_GCM_SHA256
    d. クライアントが TLS 1.2 をサポートするよう設定される場合、クライアントは以下の暗号スイートをサポートするよう設定される**べきである**：
        TLS_RSA_WITH_AES_256_GCM_SHA384

TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256
TLS_ECDHE_ECDSA_WITHAES_256_GCM_SHA384
TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256
TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256

e. クライアントは、上記およびセクション 3.3.1 または附属書 C で列挙されたもの以外の暗号スイートをサポートするよう設定**されてはならない**。
f. クライアントによって使用される暗号モジュールは、FIPS 140 認証された暗号モジュールで**なければならない**。
g. 暗号スイートで含まれるすべての暗号アルゴリズムは、乱数生成器と同様に、認証の対象範囲内で**なければならない**。
h. 乱数生成器は、NIST 暗号アルゴリズム確認プログラム(CAVP)の下で[SP800-90A]に従ってテストされ、認証されなければならす、このテストの合格結果は、暗号モジュールの FIPS140 認証書に示され**なければならない**。
i. 認証された乱数生成器は、クライアントランダム値の 28-byte ランダム値を生成するために使用され**なければならない**。
j. 認証された乱数生成器は、クライアントランダム値の 4-byte タイムスタンプを生成するために使用される**べきである**。

4. 拡張
   a. TLS クライアントは、セクション 4.4.1 で記述されるとおり、以下の TLS 拡張をサポートし**なければならない**：
      Renegotiation Indication
      Server Name Indication
   b. TLS クライアントは、セクション 4.4.2 で記述されるとおり、セクション 4.4.2 で述べられる条件が満たされるとき、以下の TLS 拡張をサポートし**なければならない**：
      Supported Elliptic Curves
      EC Point Format
      Signature Algorithms
      Certificate Status Request
      Multiple Certificate Status
   c. TLS クライアントは、セクション 4.4.2.6 で記述されるとおり、セクション 4.4.2 で述べられる条件が満たされるとき、Trusted CA Indication 拡張をサポートする**べきである**。
   d. TLS クライアントは、セクション 4.4.2.7 で記述されるとおり、セクション 4.4.2 で述べられる条件が満たされるとき、Truncated HMAC 拡張をサポートしてもよい。
   e. TLS クライアントは、以下の TLS 拡張をサポート**するべきではない**：
      Client Certificate URL
5. サーバ認証
   a. クライアントは、適切なトラストアンカがストアにある場合、クライアントトラストアンカストアにある少なくとも一つのトラストアンカを用いて TLS ハンドシェイクでのサーバ証明書の証明パスを構築することができ**なければならない**。
   b. クライアントは、証明パスを構築するために以下の資源のすべてまたはサブセットを使用することができる：ローカル証明書ストア、ハンドシェイク中にサーバから受信した証明書、LDAP, さまざまな CA 証明書における Subject Information Access 拡張の CA Repository フィールドで宣言された資源、およびさまざまな証明書における Authority Information Access 拡張の CA Issuers フィールドで宣言された資源。
   c. クライアントは、[RFC5280]のセクション 6 で規定される証明パス検証規則に従ってサーバ証明書を検証し**なければならない**。
   d. クライアントは、証明パスにおけるそれぞれの証明書の失効状態が証明書失効リスト(CRL)またはオンライン証明書状態プロトコル(OCSP)を用いてチェックされ**なければならない**。
   e. クライアントが OCSP をサポートする場合、OCSP チェックは[RFC6960]に従ったもので**なければならず**、本文書のセクション 4.5.1 で記述されたオプションの一つのみを使用する**べき**

である。
f. クライアントは、パス検証が失敗した場合、TLS コネクションを終了しなければならない。
g. クライアントは、クライアント TLS 要求にある DNS 名または IP アドレスがサーバ証明書の subject alternative name 拡張に含まれる名前または IP アドレスと一致することをチェックしなければならない。
h. クライアント TLS 要求にある名前がサーバ証明書の subject alternative name 拡張にない場合、クライアントは、サブジェクト識別名が要求された名前を含まないかどうかを決定するため、サーバ証明書の subject distinguished name フィールドをチェックしなければならない。
i. クライアントは、名前チェックが失敗した場合、TLS コネクションを終了しなければならない。
j. クライアントは、クロス証明経由で検証可能なさまざまな CA 証明書を用いてそれらのトラストストアを過密にしてはならない。
k. クライアントは、セクション 3.5.4 で議論されるとおり、過密にしているまたはヒントリストを提供しないサーバトラストストアに信頼を置かなければならない。
l. クライアントは、クライアント実装がそうするためのメカニズムを提供する場合、サーバ鍵長をチェックしなければならない。これは、サーバ証明書の公開鍵とマスタシークレットを確立するために使用されるサーバの一時的公開鍵の両方に適用可能である。

6. セッション再開
   a. それぞれのコネクションセッションについてサーバを認証する要求事項がある場合、クライアントは、先に進むためにはハンドシェイク手順全体(サーバ認証を含めて)を実行するような、新しいセッション ID を生成しなければならない。
7. 圧縮モード
   a. クライアントは、TLS 圧縮を無効化するような null 圧縮方法をサポートするべきである。
   b. 圧縮が使用される場合、クライアントは[RFC3749]で定義される方法をサポートしなければならない。
      i. 提供されるサーバ人口が[RFC3943]における圧縮方法をサポートすることが知られている場合、その方法が代わりに使用されてもよい。
   c. クライアントは、その他の圧縮方法をサポートしてはならない。

4.9.3 クライアント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管理者とは、日々の TLS クライアントを維持する責任を持つ個人である。

1. バージョンサポート
   a. システム管理者は、2015 年 1 月 1 日までに TLS 1.2 をサポートするような移行計画を策定しなければならない。
2. 証明書
   a. システム管理者は、セクション 4.9.2 の証明書推奨事項に従って証明書をインストール、維持、および更新しなければならない。
3. サーバ認証
   a. システム管理者は、さまざまなクライアントマシンにおけるトラストアンカストアを決定するために商用ウェブサイトのアクセスに関連するリスクとその必要性の間のトレードオフを実行しなければならない。
   b. システム管理者は、集中管理アプリケーションを通してトラストアンカストアを管理しなければならない。
   c. システム管理者は、トラストアンカストアへのアップデートが適切な政府機関のセキュリティ承認を要求するような特権システム管理者機能であるように、クライアントを設定しなければならない。
   d. 管理者は、クロス証明を経由して信頼されるようなさまざまな CA 証明書を用いてクライアントトラストストアが過密にされないことを保証しなければならない。
      i. 代わりに、クライアントは、過密になったサーバまたはセクション 3.5.4 で議論されたとおりのヒントリストを提供しないようなサーバに信頼を置かなければならない。

4. 運用上の検討事項
    a. クライアントと関連するプラットフォームは、セキュリティパッチに関して最新に維持されなければならない。

4.9.4 エンドユーザのための推奨事項

エンドユーザとは、TLS コネクションを確立するようなクライアントを使用する個人である。エンドユーザのための推奨事項は以下のとおりである：

1. クライアントがブラウザである場合、利用者は、TLS セッションが実行されていることを保証するためにインタフェースを検査するべきであり、また利用者が表示されたウェブサイトを訪問しようと意図したことを保証するため、ウェブサイト URL を視覚的にも検査するべきである。
2. 利用者は、URL が正当であるように見ることが可能であるが、まだ有効でないことを知っているべきである。
3. 利用者は、政府機関および管理者の指示に従って、クライアントシステムを操作しなければならない。
4. 利用者は、クライアントの外(例．PIV カード)でクライアント認証鍵を保護するための適切なポリシーと手順に従わなければならない。

附属書 A：略語

本ガイドラインで使用される頭字語と短縮語は、以下のとおり定義される。

3DES	Triple DES (TDEA)
AEAD	Authenticated Encryption with Associated Data
AES	Advanced Encryption Standard
CA	Certification Authority (認証局)
CBC	Cipher Block Chaining (暗号ブロックチェイニング)
CCM	Counter with CBC-MAC
CRL	Certificate Revocation List (証明書失効リスト)
DES	Data Encryption Standard
DH	Diffie-Hellman key exchange
DHE	Ephemeral Diffie-Hellman key exchange
DNS	Domain Name System
DNSSEC	DNS Security Extensions
DSA	Digital Signature Algorithm
DSS	Digital Signature Standard (implies DSA)
EC	Elliptic Curve
ECDHE	Ephemeral Elliptic Curve Diffie-Hellman
ECDSA	Elliptic Curve Digital Signature Algorithm
FIPS	Federal Information Processing Standard
GCM	Galois Counter Mode
IETF	Internet Engineering Task Force
MAC	Message Authentication Code
OCSP	Online Certificate Status Protocol
OID	Object Identifier
PIV	Personal Identity Verification
PKI	Public Key Infrastructure
PRF	Pseudo-random Function
PSK	Pre-shared Key
RFC	Request for Comments
SHA	Secure Hash Algorithm
SSL	Secure Sockets Layer
TLS	Transport Layer Security
URL	Uniform Resource Locator

附属書 B：暗号スイート名の解釈

暗号スイート名は、アンダースコア(即ち、「_」(訳注：下線)) によって分離された一連のニーモニックから成る。最初のニーモニックは、プロトコル名、即ち、TLSである。このセクションは、これらのガイドラインで推奨されるような暗号スイートの名前を解釈するためのガイダンスを提供する。将来の暗号スイートは、これらの表記法に従わないかもしれない。

一つまたは二つの鍵交換アルゴリズムを示すようなニーモニックがプロトコル名に続く。ニーモニックが一つのみの場合、これらのガイドラインの推奨事項に基づき、RSAまたはPSKでなければならない。一つのニーモニックRSAは、サーバ証明書の公開鍵がクライアントによってプリマスタシークレットをサーバへ送信するために使用されるべきRSA鍵配送公開鍵であることを意味する。一つのニーモニックPSKは、[RFC4279]で記述されるとおり、プリマスタシークレットが事前共有鍵と共に対象アルゴリズムのみを用いて確立されることを示す。使用するために承認された事前共有鍵暗号スイートは、附属書Cに列挙されている。プロトコル名の後に二つのニーモニックがある場合、最初の鍵交換ニーモニックは、DH、ECDH、DHE、またはECDHEであるべきである。最初の鍵交換ニーモニックがDHまたはECDHのとき、証明書のサーバの公開鍵がDHまたはECDH鍵交換のいずれかであり、また2番目のニーモニックはサーバ証明書を署名するために発行CAによって使用された署名アルゴリズムを示す。最初の鍵交換ニーモニックがDHEまたはECDHEのとき、エフェメラルな(訳注：一時的な) DHまたはECDHが鍵交換のために使用され、2番目のニーモニックがサーバの一時的公開鍵を認証するために使用されるサーバ署名公開鍵タイプ[26]を示していることを示す。

次は、用語WITHおよび対称暗号アルゴリズムと関連する利用モードのニーモニックである。

最後のニーモニックは一般に適用可能な場合[27]、HMACに使用されるハッシュアルゴリズムである。HMACが適用可能でない場合(例．AES-GCM)、暗号スイートは、TLS 1.2 RFCのリリース後に定義される、このニーモニックはPRFのハッシュアルゴリズムをあらわす。

以下の例は、暗号スイート名を解釈する方法を説明する：

- TLS_RSA_WITH_3DES_EDE_CBC_SHA：サーバはクライアントが鍵交換で使用するであろうRSA公開鍵を使用している。CA署名アルゴリズムは、規定されていない。一度ハンドシェイクが完了すると、メッセージはトリプルDESをCBCモードで使用して暗号化される。TLSバージョン1.0oyobi1.1では、SHA-1とMD5の組合せがPRFで使用され、SHA-1がメッセージ上のHMAC計算に使用される。TLS 1.2では、SHA-256がPRFのために使用され、SHA-1がメッセージ上のHMAC計算に使用される。
- TLS_DH_DSS_WITH_AES_256_CBC_SHA256：サーバは、DH証明書をしようしている。コネクションがTLSバージョン1.2を使用し、かつ署名アルゴリズム拡張がクライアントによって提供される場合、証明書は、拡張によって規定されるアルゴリズムを用いて署名される。さもなければ、証明書は、DSAを用いて署名される。一度ハンドシェイクが完了すると、メッセージ

---

26 この場合、証明書に署名するために CA によって使用される署名アルゴリズムは暗号スイートで明示されない。

27 対称暗号利用モードが認証付き暗号、即ち CCM または GCM のとき、HMAC が適用できない。それとは別に、CCM モード暗号スイートは、最後のニーモニックを規定せず、SHA-256 が PRF に使用されることを要求する。

はAES-256をCBCモードで用いて暗号化される。SHA-256がPRFとHMAC計算の両方に使用される。SHA-1以外のセキュアハッシュアルゴリズムを規定する暗号スイートは、TLS 1.2以前ではサポートされない。

- TLS_ECDHE_ECDSA_WITH_AES_256_GCM_SHA384：一時的鍵のECDHが鍵交換で使用される。サーバの一時的公開鍵がサーバのECDSA公開鍵を用いて認証される。サーバのECDSA公開鍵を認証するために使用されるCA署名アルゴリズムは規定されない。一度ハンドシェイクが完了すると、メッセージはAES-256をGCMモードで暗号化され、認証され、SHA-384がPRFのために使用される。認証付き暗号化モードが使用されるので、メッセージはHMACメッセージ認証コードを持たないし、要求もしない。

附属書 C：事前共有鍵

事前共有鍵 (PSK) は、TLSセッションの開始前にすでに設定されている対称鍵である(例．手動配付の結果として)。TLSプロトコルでのPSKの使用は、[RFC4279]、[RFC5487]、および[RFC5489]で記述されている。一般に事前共有鍵は、使用される**べきではない**。しかし、事前共有鍵の使用は、適切な鍵管理サポートのあるような、何らかの閉鎖的な環境に対して適切であるかもしれない。例えば、処理、メモリ、または電源が制限されたような制約的な環境に対しては適切かもしれない。PSKが適切であり、サポートされている場合、以下の追加のガイドラインに従わ**なければならない**。

推奨される事前共有鍵(PSK)暗号スイートは、表C-1に列挙されている；事前共有鍵は、セキュアな手動配付または鍵確立証明書を用いる等のセキュアなやり方で配付され**なければならない**。これらの暗号スイートは、エンティティ認証 (サーバとクライアントの両方) のための事前共有鍵を採用し、鍵確立のためにRSAまたは一時的な鍵のDiffie-Hellman(DHE)アルゴリズムを使用してもよい。例えば、DHEが使用されるとき、Diffie-Hellmanの計算の結果が事前共有鍵とプリマスタシークレットを決定するためのその他の入力と共に結合される。

事前共有鍵は、最小限112 bitsのセキュリティ強度を持た**なければならない**。なぜなら、暗号スイートは事前共有鍵を要求し、これらのスイートは、古典的なセキュアうぇびサイトアプリケーションに対して一般的に利用可能ではなく、またTLSクライアントまたはTLSサーバで広くサポートされていると期待されないからである。NISTはこれらのスイートは、特にネットワークエンティティの頻繁な認証が要求される場合、特に基盤アプリケーション用と考えられているとNISTは示唆している。これらの暗号スイートは、TLSバージョン1.2または1.2と共に使用されるかもしれない。GCM、SHA-256、またはSHA-384を用いる暗号スイートは、TLS 1.2でのみ利用可能であることに注すること。

事前共有鍵暗号スイートは、クライアントとサーバの両方が政府機関システムであるようなネットワークでのみ使用されてもよい。事前共有鍵を用いる暗号スイートは、TLS 1.0 がサポートされるとき、サポート**されてはならない**、またクライアントまたはサーバが非政府機関のシステムと通信するような場合サポート**されてはならない**。

表 C-1：事前共有鍵暗号スイート

Cipher Suite Name 暗号スイート名	Key Exchange 鍵交換	Encryption 暗号化	Hash function for HMAC HMAC 用のハッシュ関数	Hash Function for PRF PRF 用のハッシュ関数
TLS_PSK_WITH_3DES_EDE_CBC_SHA	PSK	3DES_EDE_CBC	SHA-1	Per RFC
TLS_PSK_WITH_AES_128_CBC_SHA	PSK	AES_128_CBC	SHA-1	Per RFC
TLS_PSK_WITH_AES_256_CBC_SHA	PSK	AES_256_CBC	SHA-1	Per RFC
TLS_PSK_WITH_AES_128_GCM_SHA256	PSK	AES_128_GCM	N/A	SHA-256
TLS_PSK_WITH_AES_256_GCM_SHA384	PSK	AES_128_GCM	N/A	SHA-384
TLS_DHE_PSK_WITH_3DEA_EDE_CBC_SHA	DHE_PSK	3DES_EDE_CBC	SHA-1	Per RFC
TLS_DHE_PSK_WITH_AES_128_CBC_SHA	DHE_PSK	AES_128_CBC	SHA-1	Per RFC
TLS_DHE_PSK_WITH_AES_256_CBC_SHA	DHE_PSK	AES_256_CBC	SHA-1	Per RFC
TLS_DHE_PSK_WITH_AES_128_GCM_SHA256	DHE_PSK	AES_128_GCM	N/A	SHA-256

TLS_DHE_PSK_WITH_AES_256_GCM_SHA384	DHE_PSK	AES_128_GCM	N/A	SHA-384
TLS_RSA_PSK_WITH_3DES_EDE_CBC_SHA	RSA_PSK	3DES_EDE_CBC	SHA-1	Per RFC
TLS_RSA_PSK_WITH_AES_128_CBC_SHA	RSA_PSK	AES_128_CBC	SHA-1	Per RFC
TLS_RSA_PSK_ WITH_AES_256_CBC_SHA	RSA_PSK	AES_256_CBC	SHA-1	Per RFC
TLS_RSA_PSK_ WITH_AES_128_GCM_SHA256	RSA_PSK	AES_128_GCM	N/A	SHA-256
TLS_RSA_PSK_WITH_AES_256_GCM_SHA384	RSA_PSK	AES_128_GCM	N/A	SHA-384
TLS_ECDHE_PSK_WITH_3DES_EDE_CBC_SHA	ECDHE_PSK	3DES_EDE_CBC	SHA-1	Per RFC
TLS_ECDHE_PSK_WITH_AES_128_CBC_SHA	ECDHE_PSK	AES_128_CBC	SHA-1	Per RFC
TLS_ECDHE_PSK_WITH_AES_256_CBC_SHA	ECDHE_PSK	AES_256_CBC	SHA-1	Per RFC
TLS_ECDHE_PSK_WITH_AES_128_GCM_SHA256	ECDHE_PSK	AES_128_GCM	SHA-256	SHA-256
TLS_ECDHE_PSK_WITH_AES_256_GCM_SHA384	ECDHE_PSK	AES_128_GCM	SHA-384	SHA-384

附属書 D：将来の機能

このセクションは、TLS に適用可能な新しい概念と機能を識別している。これらの概念が成熟するにつれ、商用製品がそれらをサポートするため利用可能となり、これらのガイドラインは、具体的な推奨事項を提供するよう改訂されるだろう。

D.1　追加の／代替のウェブサーバ証明書検証メカニズム

CA、登録システム、またはプロセスの危殆化に関連する脅威に対処するため、TLS セッションで提示されるサーバ証明書の正当性の保証を得る方法についての新しいアイディアが開発されてきた。

さらに、新しい標準が DNS をセキュアにするための公開鍵技術の使用において生み出されている。これらの DNSSEC 標準は、サーバ証明書における信頼を確立する伝統的な PKI アプローチを置き換え、または追加するために使用されることが可能である。

以下のセクションは、これらの概念を記述する。いくつかの場合において、これらの概念は十分に規格化されていない、またほとんどの場合において、それらは商用製品において幅広く利用可能ではない。これらの概念が成熟し、幅広く使用可能となるにつれて、これらのガイドラインは、それらをさらに記述するため、またそれらがサーバ証明書と関連する失効チェックにおける信頼を確立するための伝統的なメカニズムを追加しまたは置き換えるために使用されることを可能とする方法について推奨するために改訂されるだろう。

D.1.1　ソブリンキー（Sovereign Keys）

ソブリンキーアプローチは、Electronic Frontier Foundation によって開発された。このアプローチの下、サーバ公開鍵証明書および追加の中間 CA 証明書は、サーバドメインホルダーによって主張される、またこれらの主張は、一つまたは複数の信頼されるサードパーティによって裏書署名される。クライアントシステムがこれらの信頼されるサードパーティ公開鍵と共に出荷されるとき、クライアントは、レコードを問い合わせることができ、また TLS ハンドシェイクで提示されているサーバ証明書が正当である（即ち、信頼されるサードパーティにより署名される）ことを検証するための主張を得ることができる。この概念は、[SOVER]においてさらに記述される。この概念はまだ開発ステージにあるが、その仕様によって、公開鍵証明パスの作成、検討および失効チェックの必要性をなくすことができ、セクション 4.5 で列挙されたサーバ認証要求事項を置き換えることができる。

D.1.2　証明書の透明性

Google の証明書透明性プロジェクト[RFC6962] は、CA 署名された証明書の発行をより透明にすることによって証明書ベースの今日の影響を軽減しようと努力している。これは、証明書の公開ログ、公開ログモニタリング、および公開証明書監査の使用を通して行われる。証明書ログは、公開精密検査に対して公開されるような、暗号学的に保証された証明書のレコードである。証明書は、ログに追加されるかもしれないが、それらは削除、改ざん、またはログの中間へ挿入することはできない。モニターは、再提示が求められるようなドメインによって許可されなかったようなもの等、疑わしい証明書についての証明書ログを見張る。監査者は、ログの完全性と一貫性を検証するのと同様に、ログにおける特定の証明書の会員資格をチェックする能力を持つ。

D.1.3　パースペクティブズ（Perspectives）とコンバージェンス（Convergence）

パースペクティブズは、カーネギーメロン大学で行われたプロジェクトである[PERSP]。
パースペクティブズは、認証局における信頼および X.509 における公開鍵証明書信頼モデルおよび[RFC5280]を用いるよりも、TLS サーバ公開鍵証明書における信頼を確立するという異なるアプローチを取っている。パースペクティブズは、「ネットワーク公証サーバ」を用いるような分散化モデルを持っている。ネットワーク公証サーバは、インターネットに接続され、それぞれのサイトによって使用され

る TLS 証明書の履歴を構築するためにウェブサイトを常時モニターする。[RFC5280]およびセクション 4.5 で記述されるような TLS サーバ証明書を検証するよりも、パースペクティブズと共に、TLS クライアントは、時間を越えてネットワーク公証によって観測される証明書との一貫性をチェックすることによって証明書を検証する。クライアントは、自身に組み込まれたネットワーク公証の公開鍵を持っており、交渉サーバのどらが知らん出来るか、いくつが信頼できるかを決定する。クライアントは、TLS サーバ公開鍵証明書を信頼する前にいくつの公証サーバが賛成の応答をしなければならないかについても決定でき、信頼の履歴と利用者の入力を用いて決定を追加することができる。[PERSP]はパースペクティブズのさらに記述している。パースペクティブズによって使用される分散化モデルは、一つまたはいくつかの危殆化した「ネットワーク公証」に対して保護しながらも、高度の信頼性と利用可能性を提供する。パースペクティブズの実装は[Perspectives]で利用可能である。

コンバージェンス[Convergence]は、包括的なソリューションを形成するそれらのアイディアを追加するのと同様に、パースペクティブズプロジェクトからの概念を実装するための別の取組みである。特に、これは、元々のパースペクティブズの作業において存在していた完全性、プライバシーおよび即応性の問題に対処している。コンバージェンス公証は、証明書が信頼されるべきかどうかを決定するために、ネットワークパースペクティブズを越えた追加の方法を採用することもできる。

パースペクティブズ／コンバージェンスのアプローチは、自己署名された TLS サーバ証明書において信用を確立するために使用されることが可能であり、そうする際に、利用者に提示されるような証明書の警告の量を軽減する。

D.1.4 DANE

標準と製品は、名前付けされたエンティティの DNS ベース認証（DNS-based Authentication of Named Entities (DANE)の分野において、またいくつかの標準が情報提供 [RFC6394] が、まだ新しく生まれている。しかし、以下のメカニズムの一つは、TLS サーバ認証のセキュリティにおいて援助することができ、CA または登録システムとプロセスでのエラーや危殆化ゆえに発行された s 居 k されない証明書を受け入れてしまうことからのクライアントを保護することができる：

1. セクション 4.5 で規定されるとおりサーバ公開鍵証明書検証に追加して、クライアントは、TLS サーバ証明書が DNS レコードにおいて提供されるものと一致することを検証する。DNS レコード上のディジタル署名は、[RFC4033]で記述されるとおり、DNS セキュリティ拡張(DNSSEC)に従って検証される。

2. クライアントは、セクション 4.5 できていされるとおり、サーバ公開鍵証明書検証を見合わせる。その代わり、クライアントは、TLS サーバ証明書が DNS レコードにて提供するものと一致することを検証する。DNS レコード上のディジタル署名は、[RFC4033]で記述されるとおり、DNS セキュリティ拡張(DNSSEC)に従って検証される。

3. セクション 4.5 で規定された通り、サーバ公開鍵証明書検証に追加して、クライアントはハンドシェイク中にサーバによって提供された証明書リストにある CA 証明書が DNS レコードで提供され、またセクション 4.5 で規定されるとおり検証された証明パスの一部である証明書と一致することを検証する。DNS レコード上のディジタル署名は、[RFC4033]で記述されるとおり、DNS セキュリティ拡張(DNSSEC)に従って検証される。

4. クライアントは、TLS サーバ証明書が DNS レコードで提供されるトラストアンカによって検

証されることが可能であることを検証する。DNS レコード上のディジタル署名は、[RFC4033]
でき樹うつぁれるとおり、DNS セキュリティ管区長(DNSSEC)に従って検証される。

D.2 サーバ／クライアント鍵長のチェック

クライアントまたはサーバが特定の鍵長またはアルゴリズムを要求しようと望む場合、それらは[RFC5698]で定義される概念を用いる暗号あるごり z 無ポリシーを実装することができる。[RFC5698]で記述されるとおり、暗号アルゴリズムポリシーの仕様と処理は、TLS ハンドシェイクにおける暗号スイート仕様にかかわらず、受入できないアルゴリズムと鍵長が暗号アルゴリズムポリシーを実装したエンティティ(クライアントまたはサーバ)によって受け入れられないことを保証することができる。

D.3 Encrypt-then-MAC 拡張

TLS ワーキンググループは、一つの拡張 [ETM] として、TLS への Encrypt-then-MAC 構築の追加に向けて作業中である。これは、[RFC2246]、[RFC4346]、および[RFC5246]で規定された MAC-then-Encrypt 構築からの出発である。Encrypt-then-MAC 拡張が標準化される場合、CBC 山号スイート上のいくつかの既知の攻撃を軽減または防止するだろう。

附属書 E　　参考文献

以下のリストの文書、出版、および組織は、トランスポート層セキュリティのさまざまな観点における幅広いさまざまな情報を提供する。

[Adams99] Adams, C. and Lloyd, S., *Understanding PKI: Concepts, Standard, and Deployment Considerations*, (Macmillan Technology Publishing, Indianapolis, IN, ISBN 1-57870-166-X, 1999).

[CABBASE] *Baseline Requirements for the Issuance and Management of Publicly-Trusted Certificates*, CA Browser Forum, Version 1.1.6, 29 July 2013. https://cabforum.org/wp-content/uploads/Baseline_Requirements_V1_1_6.pdf

[Comer00] Comer, D. E., *Internetworking with TCP/IP, Principles, Protocols, and Architectures*, Fourth Edition, (Prentice Hall, Upper Saddle River, NJ 07458, ISBN: 0-13- 018380-6, 2000).

[COMMON] X.509 Certificate Policy for the U.S. Federal PKI Common Policy Framework, Version 1.21, 18 December 2012. http://idmanagement.gov/documents/common-policy-framework-certificate-policy

[Convergence] Thoughtcrime Labs, *Convergence*, http://convergence.io/

[ETM] Gutmann, P., *Encrypt-then-MAC for TLS and DTLS*, Internet Engineering Task Force, December 2013, http://tools.ietf.org/html/draft-gutmann-tls-encrypt-then-mac-05

[EVGUIDE] *Guidelines For The Issuance and Management of Extended Validation Certificates*, CA Browser Forum, Version 1.4.3, 9 July 2013. https://cabforum.org/wp-content/uploads/Guidelines_v1_4_3.pdf

[FBCACP] X.509 Certificate Policy for the Federal Bridge Certification Authority, Version 2.26, 26 April 2012. http://www.idmanagement.gov/sites/default/files/documents/FBCA%20Certificate%20Policy%20v2.26_s.pdf

[FIPS140-2] FIPS 140-2, *Security Requirements For Cryptographic Modules*, http://csrc.nist.gov/publications/fips/fips140-2/fips1402.pdf

[FIPS140Impl] National Institute of Standards and Technology, *Implementation Guidance for FIPS PUB 140-2 and the Cryptographic Module Validation Program*, 25 July 2013, http://csrc.nist.gov/groups/STM/cmvp/documents/fips140-2/FIPS1402IG.pdf

[FIPS180-4] National Institute of Standards and Technology, *Secure Hash Standard*, Federal Information Processing Standards Publication 180-4, March 2012, http://csrc.nist.gov/publications/fips/fips180-4/fips-180-4.pdf

[FIPS186-4] National Institute of Standards and Technology, *Digital Signature Standard*, Federal Information Processing Standard 186-4, July 2013, http://nvlpubs.nist.gov/nistpubs/FIPS/NIST.FIPS.186-4.pdf

[FIPS197] National Institute of Standards and Technology, *Advanced Encryption Standard (AES)*, Federal Information Processing Standard 197, November 26, 2001 http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf

[FIPS198-1] National Institute of Standards and Technology, *The Keyed-Hash Message Authentication Code (HMAC)*, Federal Information Processing Standard 198-1, July 2008, http://csrc.nist.gov/publications/fips/fips198-1/FIPS-198-1_final.pdf

[FIPS201-1] National Institute of Standards and Technology, *Personal Identification Verification (PIV) of Federal Employees and Contractors*, Federal Information Processing Standard 201-1, March 2006, http://csrc.nist.gov/publications/fips/fips201-1/FIPS-201-1-chng1.pdf

[Hall00] Hall, E. A., *Internet Core Protocols, The Definitive Guide*, (O'Reilly & Associates, ISBN: 1-56592-572-6, February 2000).

[Housley01] Housley, R. and Polk, T., *Planning for PKI, Best Practices Guide for Deploying Public Key Infrastructure*, (John Wiley & Sons, New York, NY, ISBN 0-471-39702-4, 2001).

[Lucky13] AlFardan, N. J., and Paterson, K. G., *Lucky Thirteen: Breaking the TLS and DTLS Record Protocols*, IEEE Symposium on Security and Privacy 2013, pages 526-540, full version at http://www.isg.rhul.ac.uk/tls/TLStiming.pdf

[Paterson11] Paterson, K. G., Ristenpart, T., and Shrimpton, T., *Tag Size* Does *Matter: Attacks and Proofs for the TLS Record Protocol*, in ASIACRYPT 2011, (Springer Lecture Notes in Computer Science, volume 7073, ISBN 978-3-642-25384-3).

[PERSP] Wendlandt D., Andersen D.G. and Perrig A., *Perspectives: Improving SSH-style Host Authentication with Multi-Path Probing*, 2011 USENIX. http://perspectivessecurity.files.wordpress.com/2011/07/perspectives_usenix08.pdf.

[Perspectives] Perspectives Project, http://perspectives-project.org/

[Polk03] Polk, W., Hastings, N., and Malani, A., *Public Key Infrastructures that Satisfy Security Goals*, IEEE Internet Computing, Volume 7, Number 4, July-August, 2003.

[Rescorla01] Rescorla, E., *SSL and TLS – Designing and Building Secure Systems*, (Addison-Wesley, Upper Saddle River NJ, 07458, ISBN 0-201-61598, March 2001).

[RFC2119] Bradner, S., *Key words for use in RFCs to Indicate Requirement Levels*, Internet Engineering Task Force, Request for Comments 2119, March 1997, http://www.ietf.org/rfc/rfc2119.txt

[RFC2246] Dierks, T. and Allen, C., *The TLS Protocol Version 1.0*, Internet Engineering Task Force, Request for Comments 2246, January 1999, http://www.ietf.org/rfc/rfc2246.txt

[RFC3279] Polk, W., et al., *Algorithms and Identifiers for the Internet X.509 Public Key Infrastructure Certificate Revocation List (CRL) Profile*, Internet Engineering Task Force, Request for Comments 3279, April 2002, http://www.ietf.org/rfc/rfc3279.txt

[RFC3447] Jonsson, J., and Kaliski, B., *Public-Key Cryptography Standards (PKCS) #1: RSA Cryptography Specifications Version 2.1*, Request for Comments 3447, February 2003, http://www.ietf.org/rfc/rfc3447.txt

[RFC3713] Matsui, M., et al. *A Description of the Camellia Encryption Algorithm*, Internet Engineering Task Force, Request for Comments 3713, April 2004, http://www.ietf.org/rfc/rfc3713.txt

[RFC3749] Hollenbeck, S., *Transport Layer Security Protocol Compression Methods*, Internet Engineering Task Force, Request for Comments 3749, May 2004, http://www.ietf.org/rfc/rfc3749.txt

[RFC3943] Friend, R., *Transport Layer Security (TLS) Protocol Compression Using Lempel-Ziv-Stac (LZS)*, Internet Engineering Task Force, Request for Comments 3943, November 2004, http://www.ietf.org/rfc/rfc3943.txt

[RFC4033] Arends, R. et al., *DNS Security Introduction and Requirements*, Internet Engineering Task Force, Request for Comments 4033, March 2005, http://www.ietf.org/rfc/rfc4033.txt

[RFC4055] Shaad, J. et al., *Additional Algorithms and Identifiers for RSA Cryptography for use in the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Internet Engineering Task Force, Request for Comments 4055, June 2005, http://www.ietf.org/rfc/rfc4055.txt

[RFC4279] Eronen, P. and Tschofenig, H. *Pre-Shared Key Ciphersuites for Transport Layer Security (TLS)*, Internet Engineering Task Force, Request for Comments 4279, December 2005, http://www.ietf.org/rfc/rfc4279.txt

[RFC4346] Dierks, T. and Rescorla, E., *The Transport Layer Security (TLS) Protocol Version 1.1*, Internet Engineering Task Force, Request for Comments 4346, April 2006, http://www.ietf.org/rfc/rfc4346.txt

[RFC4492] Blake-Wilson, S., et al., *Elliptic Curve Cryptography (ECC) Cipher Suites for Transport Layer Security (TLS)*, Internet Engineering Task Force, Request for Comments 4492, May 2006, http://www.ietf.org/rfc/rfc4492.txt

[RFC5246] Dierks, T. and Rescorla, E., *The Transport Layer Security (TLS) Protocol Version 1.2*, Internet Engineering Task Force, Request for Comments 5246, August 2008, http://www.ietf.org/rfc/rfc5246.txt

[RFC5280] Cooper, D., et al., *Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Internet Engineering Task Force, Request for Comments 5280, May 2008, http://www.ietf.org/rfc/rfc5280.txt

[RFC5288] Salowey, J., Choudhury, A., and McGrew, D., *AES Galois Counter Mode (GCM) Cipher Suites for TLS*, Internet Engineering Task Force, Request for Comments 5288, August 2008, http://www.ietf.org/rfc/rfc5288.txt

[RFC5289] Rescorla, E., *TLS Elliptic Curve Cipher Suites with SHA-256/384 and AES Galois Counter Mode (GCM)*, Internet Engineering Task Force, Request for Comments 5289, August 2008, http://www.ietf.org/rfc/rfc5289.txt

[RFC5487] Badra, M., *Pre-Shared Key Cipher Suites for TLS with SHA-256/384 and AES Galois Counter Mode*, Internet Engineering Task Force, Request for Comments 5487, March 2009, http://www.ietf.org/rfc/rfc5487.txt

[RFC5489] Badra, M. and Hajjeh, I., *ECDHE_PSK Cipher Suites for Transport Layer Security (TLS)*, Internet Engineering Task Force, Request for Comments 5489, March 2009, http://www.ietf.org/rfc/rfc5489.txt

[RFC5698] Kunz, T., Okunick, S., and Pordesch U., *Data Structure for the Security Suitability of Cryptographic Algorithms (DSSC)*, Internet Engineering Task Force, Request for Comments 5698, November 2009, http://www.ietf.org/rfc/rfc5698.txt

[RFC5746] Rescorla E. et al., *Transport Layer Security (TLS) Renegotiation Indication Extension*, Internet Engineering Task Force, Request for Comments 5746, February 2010, http://www.ietf.org/rfc/rfc5746.txt

[RFC5758] Dang, Q., et al., *Internet X.509 Public Key Infrastructure: Additional Algorithms and Identifiers for DSA and ECDSA*, Internet Engineering Task Force, Request for Comments 5758, January 2010, http://www.ietf.org/rfc/rfc5758.txt

[RFC6066] Eastlake, D., *Transport Layer Security (TLS) Extensions: Extension Definitions*, Internet Engineering Task Force, Request for Comments 6066, January 2011, http://www.ietf.org/rfc/rfc6066.txt

[RFC6101] Freier, A. e al., *The Secure Sockets Layer (SSL) Protocol Version 3.0*, Internet Engineering Task Force, Request for Comments 6101, August 2011, http://www.ietf.org/rfc/rfc6101.txt

[RFC6394] Barnes, R., *Use Cases and Requirements for DNS-Based Authentication of Named Entities (DANE)*, Internet Engineering Task Force, Request for Comments 6394, October 2011, http://www.ietf.org/rfc/rfc6394.txt

[RFC6460] Salter, M. and Housley, R., *Suite B Profile for Transport Layer Security (TLS)*, Internet Engineering Task Force, Request for Comments 6460, January 2012, http://www.ietf.org/rfc/rfc6460.txt

[RFC6655] McGrew, D. and Bailey, D., *AES-CCM Cipher Suites for Transport Layer Security (TLS)*, Internet Engineering Task Force, Request for Comments 6655, July 2012, http://www.ietf.org/rfc/rfc6655.txt

[RFC6818] Yee, P., *Updates to the Internet X.509 Public Key Infrastructure Certificate and Certificate Revocation List (CRL) Profile*, Internet Engineering Task Force, Request for Comments 6818, January 2013, http://www.ietf.org/rfc/rfc6818.txt

[RFC6960] Santesson, S., et al., *X.509 Internet Public Key Infrastructure Online Certificate Status Protocol - OCSP*, Internet Engineering Task Force, Request for Comments 6960, June 2013, http://www.ietf.org/rfc/rfc6960.txt

[RFC6961] Pettersen, Y., *The Transport Layer Security (TLS) Multiple Certificate Status Request Extension*, Internet Engineering Task Force, Request for Comments 6961, June 2013, http://www.ietf.org/rfc/rfc6961.txt

[RFC6962] Laurie, B., et al., *Certificate Transparency*, Internet Engineering Task Force, Request for Comments 6962, June 2013, http://www.ietf.org/rfc/rfc6962.txt

[SOVER] *Sovereign Key Cryptography for Internet Domains*, Electronic Frontier Foundation, https://git.eff.org/?p=sovereign-keys.git;a=blob_plain;f=sovereign-key-design.txt;hb=master

[SP800-32] NIST Special Publication 800-32, *Introduction to Public Key Technology and the Federal PKI Infrastructure*, February 2001, http://csrc.nist.gov/publications/nistpubs/800-32/sp800-32.pdf

[SP800-53] NIST Special Publication 800-53, *Security and Privacy Controls for Federal Information Systems and Organizations*, April 2013, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-53r4.pdf

[SP800-56A] NIST Special Publication 800-56A, *Recommendation for Pair-Wise Key Establishment Schemes Using Discrete Logarithm Cryptography*, May 2013, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-56Ar2.pdf

[SP800-56B] NIST Special Publication 800-56B, *Recommendation for Pair-Wise Key Establishment Schemes Using Integer Factorization Cryptography*, August 2009, http://csrc.nist.gov/publications/nistpubs/800-56B/sp800-56B.pdf

[SP800-57p1] NIST Special Publication 800-57 Part 1, *Recommendation for Key Management – Part 1: General (Revision 3)*, July 2012, http://csrc.nist.gov/publications/nistpubs/800-57/sp800-57_part1_rev3_general.pdf

[SP800-63] NIST Special Publication 800-63-2, *Electronic Authentication Guide*, August 2013, http://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.800-63-2.pdf

[SP800-67] NIST Special Publication 800-67 Revision 1, *Recommendation for the Triple Data Encryption Algorithm (TDEA) Block Cipher*, January 2012, http://csrc.nist.gov/publications/nistpubs/800-67-Rev1/SP-800-67-Rev1.pdf

[SP800-90A] NIST Special Publication 800-90A Revision 1, *Recommendation for Random Number Generation Using Deterministic Random Bit Generators*, January 2012, http://csrc.nist.gov/publications/nistpubs/800-90A/SP800-90A.pdf

[SP800-107] NIST Special Publication 800-107 Revision 1, *Recommendation for Applications Using Approved Hash Algorithms*, August 2012, http://csrc.nist.gov/publications/nistpubs/800-107-rev1/sp800-107-rev1.pdf

[SP800-131A] NIST Special Publication 800-131A, *Transitions: Recommendation for Transitioning the Use of Cryptographic Algorithms and Key Lengths*, January 2011, http://csrc.nist.gov/publications/nistpubs/800-131A/sp800-131A.pdf

[SP800-135] NIST Special Publication 800-135 Revision 1, *Recommendation for Existing Application-Specific Key Derivation Functions*, December 2011, http://csrc.nist.gov/publications/nistpubs/800-135-rev1/sp800-135-rev1.pdf

[Schneier96] Schneier, B., *Applied Cryptography: Protocols, Algorithms, and Source Code in C, 2nd ed.*, (John Wiley & Sons, Inc. 1996).